N. 39/1992
Sped. in abb. post. gr. III/70
L. 15.000
68000
AMIGA
BYTE
by Elettronica 2000
SUL DISCO
VIRUS CHECKER 6.01 POTENTISSIMO ANTIVIRUS
PICTURESQUE PROGRAMMA DI GRAFICA PITTORICA
POING UN'ORIGINALE VARIANTE DI ARKANOID
SHORTCUT AGGIUNGI MENU AL WORKBENCH
PRINTICON PER STAMPARE TESTI FACILMENTE
TIV VISUALIZZA/SUONA FILE IFF
BRIDGE IL CLASSICO GIOCO DI CARTE
FASTCAT UNISCI DUE AMIGA IN RETE
BUGBLASTER STERMINA GLI INSETTI
Speciale
corso di
AREXX
Spreadsheet
PROFESSIONAL CALC
TIPS & TRICKS
Tools
PROTEXT 5.5
SUPERBASE PRO. 4
M. EDIT ACTION 1.1
Hardware
TUTTO
SUGLI HARD DISK
Grafica
PONGO 1.0
TRASFORMAZIONI
TRIDIMENSIONALI
ARexx
GIOCHIAMO A DADI
CON AREXX
Amos
PRIMI PASSI
CON LA GRAFICA
Add-on
SCHEDA
ACCELERATRICE
MICROBOTICS VXL-30
Pd
CUBULUS
AMIDOCK
BUG BASH
LANDSCAPE
NEWSFLASH 23
UGA POWERFONTS 1
AMIGA CODERS CLUB 19
SOFTWARE EXPRESS

AMIGA BYTE

Direttore
SIRA ROCCHI

Direzione Editoriale
MARIO MAGRONE

Direzione Tecnica
GIANCARLO CAIRELLA

Segreteria di Redazione
SILVIA MAIER

Grafica
NADIA MARINI

Fotografie
MARIUS LOOK

Disco a cura di
VITTORIO FERRAGUTI

Copertina
GIGI MURATORE

Collaborano ad AmigaByte: Francesco Annoni, Luca Arienti, Paolo Bozzo, Luca Brigatti, Marco Brovelli, Guglielmo Cancelli, Antonio De Lorenzo, Enrico Donna, Silvia Malaguti, Vincenzo Marangoni, Luca Mirabelli, Pierluigi Montanari, Lorenzo Orlandini, Domenico Pavone, Riccardo Premoli, Guido Quaroni, Giuseppe Sacchi, Emanuele Scribanti, Paolo Sisti, Leonardo Tennozio, Aurora Tragara, Vertigo.

Redazione
C.so Vitt. Emanuele 15
20122 Milano
tel. 02/795047
ore pomeridiane.
Per telefonate tecniche: solo mercoledì h 15-18.

Amministrazione, Redazione, Pubblicità: L'Agorà srl: C.so Vittorio Emanuele 15, 20122 Milano. Fotocomposizione e fotolito: Compostudio Est., Cernusco S/N - Stampa: Arti Grafiche Gajani, Rozzano (MI). Distribuzione: SO.DI.P. Angelo Patuzzi spa, Via Zuretti 25, Milano. Amiga Byte è un periodico mensile registrato presso il Tribunale di Milano al n. 215 il 29 marzo 1988. Resp. Sira Rocchi. Spedizione in abbonamento postale Gr. III/70. Pubblicità inferiore al 70%.

IL MENU

Professional Calc

La Gold Disk ha immesso sul mercato l'ennesimo programma destinato a far parlare di sé: un foglio di calcolo potente ed intuitivo, corredato da una ricca dotazione di grafici

di LUCA MIRABELLI

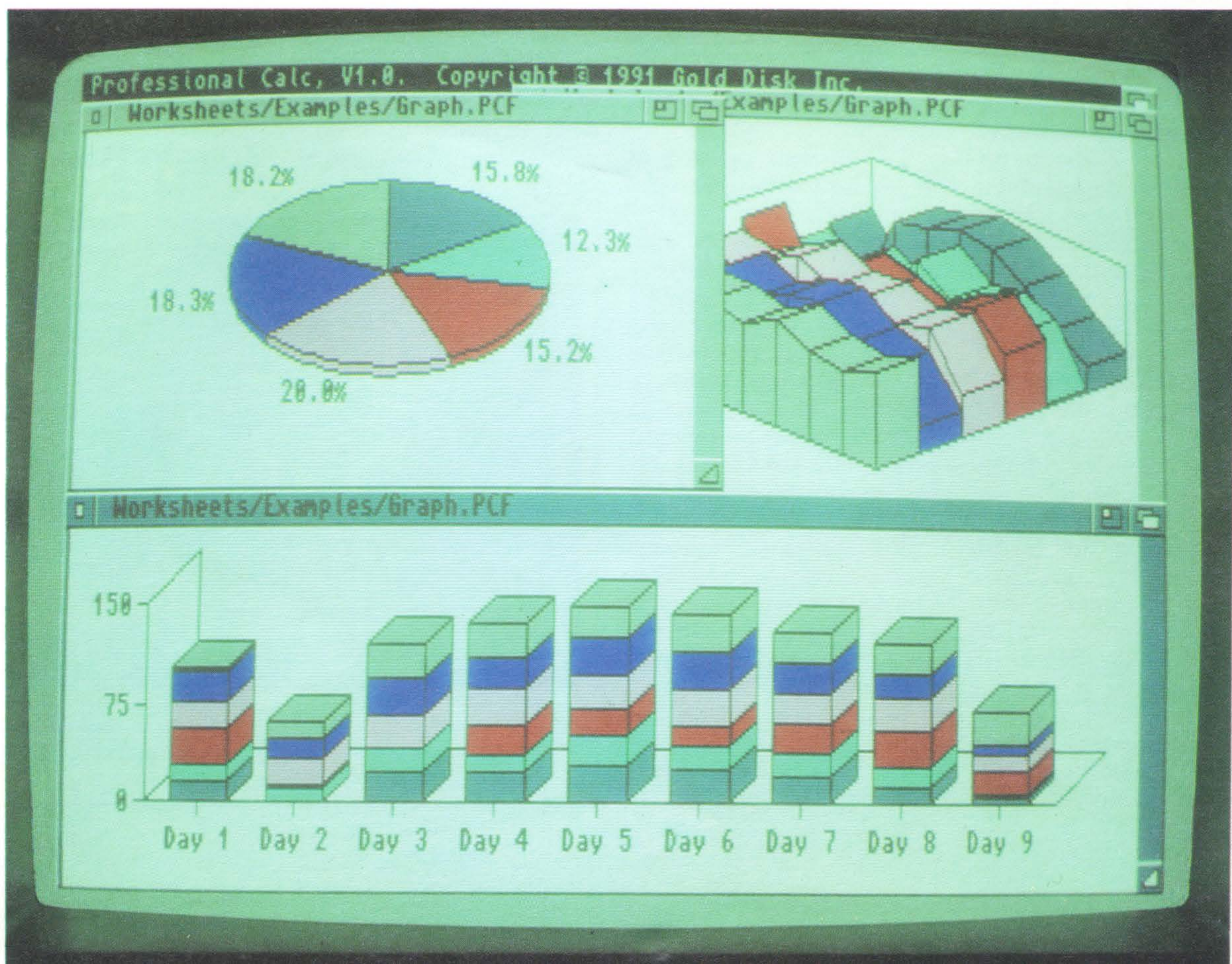
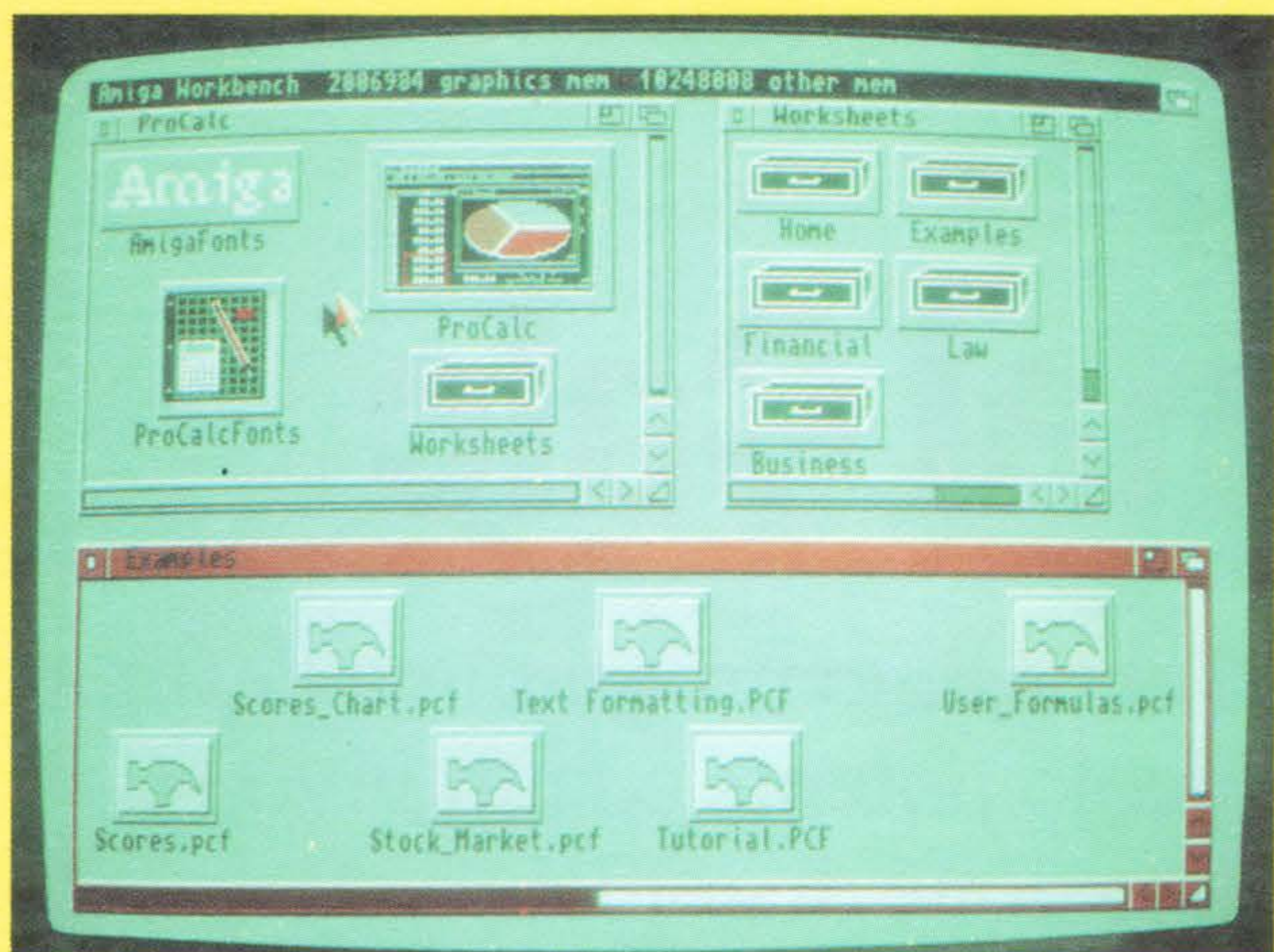
Il settore dei fogli di calcolo è probabilmente quello meno sviluppato per quanto riguarda Amiga. I programmi finora apparsi non hanno suscitato particolare entusiasmo poiché, salvo pochissime eccezioni, non sfruttavano le caratteristiche del sistema sul quale giravano, limitandosi a fornire (spesso con una lentezza mortale, e con un'interfaccia esclusivamente testuale) le basilari funzioni di calcolo.

«Professional Calc» arriva dalla Gold Disk, una casa che ha dato e continua a dare un grosso contributo nello sviluppo del software applicativo su Amiga: basti pensare a «Professional Page», a «Professional Draw» ed al recentissimo «Video Director».

Come risulta chiaro fin dalla prima schermata, l'ispirazione viene da **«Excel 3.0»**, il notissimo foglio di calcolo della Microsoft che gira in ambiente Windows (e probabilmente il più potente spreadsheet in assoluto).

Non si tratta però di un clone: i menu e le caratteristiche sono molto differenti da quelli del modello.

Apprestandoci ad entrare nel quinto anno di vita della nostra rivista, confidiamo che i lettori abbiano acquisito i principi generali del funzionamento di un foglio di calcolo, che riassumiamo in un riquadro a parte; concentreremo dunque la nostra attenzione su ciò che rende «Professional Calc» lo strumento più potente attualmente disponibile per il nostro computer.

Non ci sogneremmo mai di valutare la potenza di un programma in proporzione al numero di comandi nei menu, ma riteniamo ugualmente opportuno segnala-

Sul secondo disco si trova un discreto numero di fogli di calcolo già pronti per le applicazioni più svariate.

re che soltanto nel foglio di calcolo ce ne sono ben centotrentasei, suddivisi in otto gruppi fondamentali.

FUNZIONI PER TUTTI I GUSTI

Il primo menu, che come da tradizione è denominato **Project** e contiene i comandi relativi alla lettura/scrittura dei file, rivela le prime interessanti caratteristiche del programma: innanzitutto la possibilità di leggere e di scrivere tranquillamente i file in formato **Lotus 1-2-3** (avrete notato che la compatibilità con i principali pacchetti Ms-Dos è un aspetto sempre più diffuso nel software applicativo più recente), e di leggere i file generati da Maxiplan; inoltre, «Professional Calc» consente di tenere aperti più documenti contemporaneamente, o anche più finestre indipendenti sullo stesso documento (comando **View**).

La stampa del lavoro può avvenire mediante qualsiasi stampante supportata da Preferences, ma anche su di un dispositivo Postscript, sia in bianco e nero che a colori.

È supportato persino l'output Encapsulated PS, per l'inclusione dei vostri lavori (e in particolare dei grafici) in documenti realizzati con pacchetti di DTP.

Ulteriori comandi di I/O sono quelli relativi ai range, blocchi rettangolari di celle del foglio di calcolo. Un range può essere salvato come file Procalc, oppure in un formato ASCII, delimitato da virgole e virgolette, facilmente importabile in qualsiasi database; può anche essere trasferito nella clipboard di sistema, il metodo standard di scambio-dati tra applicazioni.

Le normali operazioni di cut & paste possono avvenire in tre modi differenti: **Identical** produce una copia conforme all'originale, **Results Only** ricopia i valori contenuti nelle celle ma non le eventuali formule che li hanno generati, **Relative** ricopia anche le formule, ma aggiornando tutti i riferimenti in rapporto alle coordinate di destinazione.

Fill è un modo molto rapido per riempire senza fatica celle adiacenti con uno stesso valore, che dovrà essere presente nella prima riga o nella prima colonna del range selezionato; **Paste Function** visualizza l'elenco delle funzioni riconosciute dal programma (per gli amanti dei numeri, sono 130: torneremo comunque più avanti su questo argomento), permettendone l'immediato inserimento nelle nostre formule.

I formati numerici riconosciuti sono sette: **Generic**; **Scientific** (in notazione esponenziale); **Percent** (sotto forma di percentuale); **Currency** (con l'indicazione della valuta); **Date** (in forma di data, espressa come numero di giorni trascorsi dal primo gennaio 1978; non sono riconosciute date precedenti); **Time** (espressa come parte della giornata già trascorsa: 0.5 indica il mezzogiorno); **Boolean**, che rappresenta lo zero con la scritta **False** e l'uno con la parola **True**.

Nei formati dove la cosa ha senso, si possono inserire automaticamente delle virgole (**Commas**) per separare le migliaia, un'opzione di nessuna utilità per i non anglosassoni, che a questo scopo usano il punto, riconosciuto invece dal programma come separatore dei decimali dalle unità.

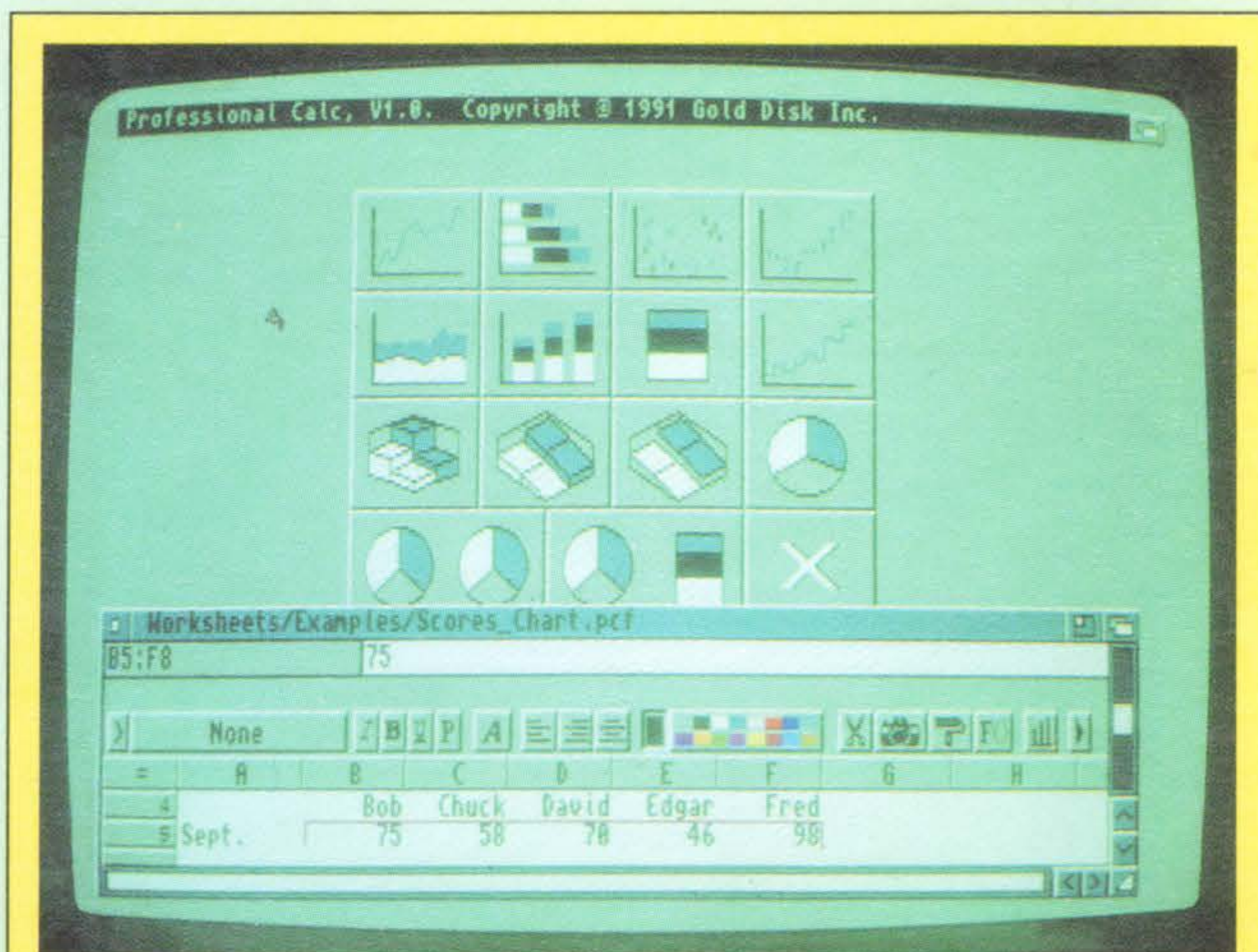
Attivando **Fixed** il numero di decimali sarà sempre pari a quello impostato; il programma provvederà, se il caso, ad aggiungere zeri per un perfetto incolonnamento dei numeri. Infine, valori negativi possono essere rappresentati dal tradizionale segno '—' oppure racchiusi tra parentesi, come è d'uso in bilanci ed in documenti analoghi.

Non è certo poco, ma siamo ben lontani dalle possibilità offerte da «Excel» in questo campo (perdonateci i continui confronti, ma in un certo senso i programmatori se la sono proprio cercata!).

PRESENTAZIONE DEI TESTI

Il contenuto di ogni cella può essere allineato a destra, a sinistra, o centrato: purtroppo la scritta non può oltrepassare il bordo sinistro della cella, e non è dunque possibile centrare testi più lunghi della cella (nessun problema, invece, per farli estendere sulle celle a destra).

I comandi **Set Color** e **Set Font** non dovrebbero destare perplessità per un utente Amiga; segnaliamo che i font contenuti sul dischetto di «Professional Calc» sono i medesimi di «Pro Page», con dimensioni da 9 a 24 punti.

Merita una citazione a parte, invece, la gestione degli **Style Tags**. Ogni Tag,

Panoramica dei tipi di grafico supportati da «Professional Calc»: quasi tutti hanno un corrispettivo tridimensionale.

UN FOGLIO MOLTO PARTICOLARE

Gli spreadsheet sono l'applicazione che, insieme alla videoscrittura, ha maggiormente contribuito alla diffusione capillare dei personal computer in tutti gli uffici.
Si tratta di piani di lavoro suddivisi in caselle rettangolari; ogni casella può contenere un numero, una scritta, oppure (è qui il bello) una formula che utilizza dati presenti in altre celle. In questo caso la formula sarà memorizzata, ma soltanto il risultato sarà visibile (e stampabile). Questo risultato può a sua volta essere richiamato da altre formule presenti in altre celle, e così via senza particolari limitazioni.

Un'accurata scelta dei font è fondamentale per conferire al vostro documento un aspetto ordinato ed accattivante.

Ogni volta che si modifica un valore, viene eseguito il cosiddetto «ricalcolo», che aggiorna tutti i valori delle celle contenenti formule sulla base dei nuovi dati.
Il foglio di calcolo si rivela quindi uno strumento potentissimo per l'elaborazione dei dati: una volta impostate le relazioni tra le grandezze in gioco nella situazione che si vuole prendere in esame (può trattarsi di un bilancio aziendale o di una simulazione scientifica), se ne può variare una ed osservare immediatamente le ripercusioni sui risultati finali.
Alcuni spreadsheet permettono anche l'operazione contraria: impostando il risultato desiderato, possono ricercare i dati di partenza che meglio lo approssimano. Contrariamente a quanto si potrebbe pensare, il Problem Solving (è questo il nome della procedura) non è affatto una cosa semplice, ed implica l'utilizzo di raffinate tecniche matematiche: probabilmente è questo il motivo per cui su Amiga non c'è ancora alcun programma che lo implementi.
La maggior parte dei pacchetti in commercio può generare grafici bi- e tri-dimensionali partendo da dati contenuti nel foglio attualmente in memoria: «Professional Calc» non fa eccezione, ed ha in questo aspetto uno dei suoi punti di forza.

contraddistinto da un nome, non è che l'insieme di tutti i parametri di formato visti finora (numeri, allineamento, tipo di carattere...), che viene memorizzato e può essere richiamato istantaneamente, risparmiando così all'utente l'impostazione manuale dei parametri suddetti.

Selezioniamo **New** per definire un nuovo tag con i parametri attualmente selezionati, ed **Apply** per richiamare uno.

Nel menu **Options** sono contenute le opzioni, per così dire, «cosmetiche»: con **Column Width** possiamo modificare la larghezza delle colonne (l'altezza delle righe è purtroppo fissa, e pari al corpo del più grande font utilizzato su ciascuna di esse); **Freeze** prende la riga (o la colonna) sulla quale si trova il cursore e la visualizza sempre nel margine della finestra. In questo modo, se dobbiamo inserire una tabella di dati piuttosto lunga, che ha sulla prima riga i nomi delle voci in essa contenute, possiamo bloccare questa riga perché rimanga visibile anche quando, continuando l'inserimento dei dati, il cursore si sposterà verso il basso. Se avete afferrato il meccanismo, non avrete difficoltà a comprendere la funzione di **Unfreeze**.

COMPLETAMENTE CONFIGURABILE

Palette e **Preferences** danno accesso ai due riquadri omonimi: per mezzo del secondo si possono definire parametri più o meno importanti quali il tipo di cursore, la presenza a video della griglia che separa le celle, le modalità di ricalcolo, l'esecuzione automatica degli script Arexx eventualmente associati alle celle (si tratta di una caratteristica davvero unica, sulla quale torneremo più avanti), l'attivazione dell'output vocale dei valori inseriti (utile, ad esempio, per gli ipovedenti).

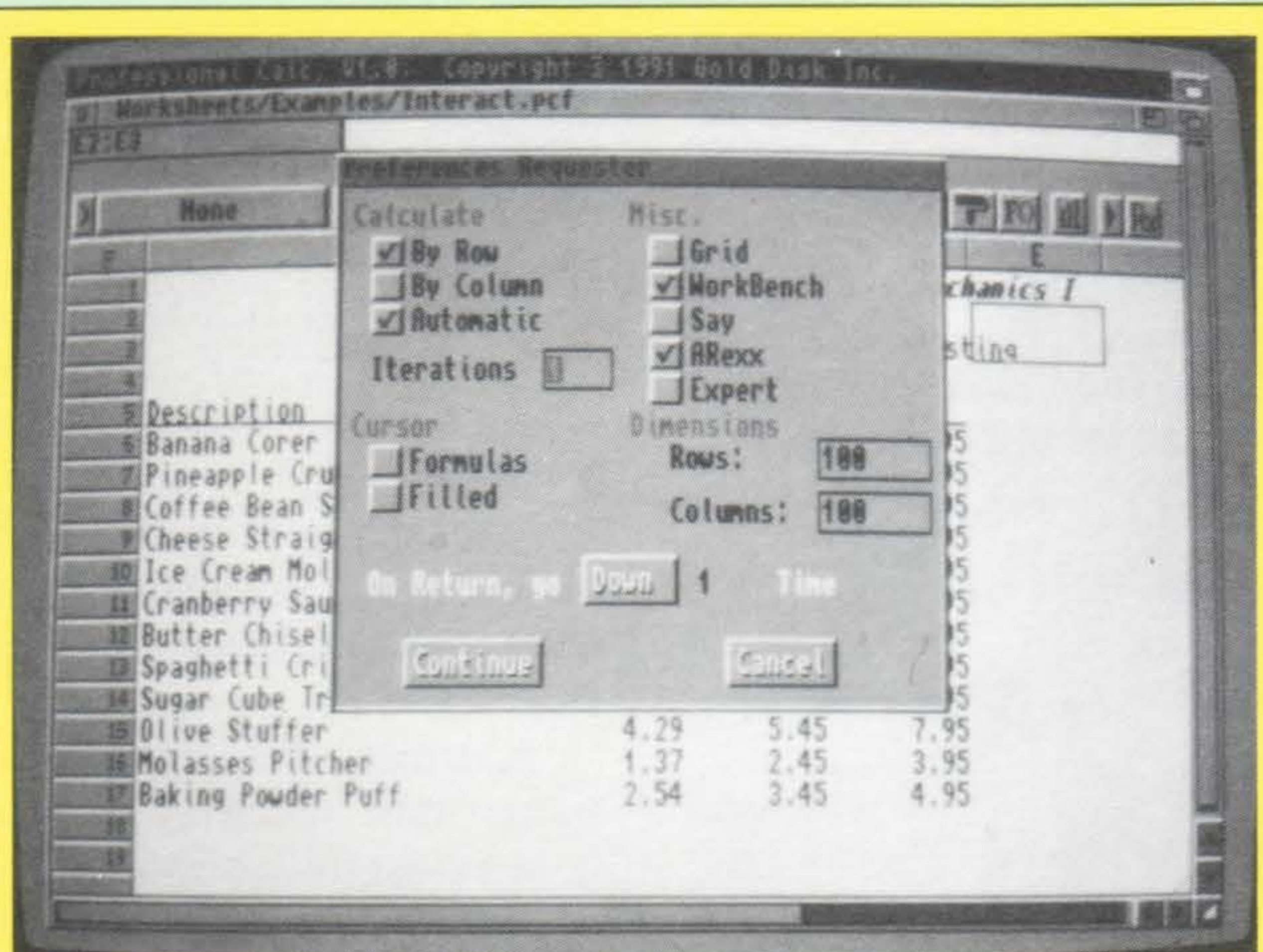
Persino la funzione del tasto RETURN è programmabile: oltre che confermare l'input in corso, può spostare il cursore di un numero qualsiasi di caselle nella direzione preferita.

A questo proposito, notiamo con rammarico che non è possibile confermare gli inserimenti con i tasti cursore.

Il menu **Commands** contiene parecchie funzioni per la gestione delle celle; innanzitutto, il ricalcolo di tutti i valori (che di default è automatico, ed i cui parametri vengono stabiliti nel riquadro **Preferences**).

Inoltre, spesso accade di dover contrassegnare una cella per distinguerla dalle altre, magari nell'ambito di un foglio molto complesso: possiamo semplicemente attribuirle un nome con **Add Name** (in seguito ci porteremo su di essa usando **Select Name**) oppure associarle una linea di commento che verrà visualizzata nella parte alta dello schermo ogni volta che la cella verrà selezionata. Se proprio il testo non può stare su di una sola linea, con **Extended Cell Note** collegheremo la cella ad un file ASCII di lunghezza a piacere, per la visualizzazione del quale è però necessario possedere «**Transwrite**», il word processor della Gold Disk, fornito con il pacchetto di «Professional Page» ma non con quello di «Professional Calc».

Prima di abbandonare questo menu notiamo **Optimize**, che riorganizza la sistemazione dei dati in modo da impegnare la minor quantità di memoria possibile.

Outline viene in aiuto quando occorre nascondere alla vista (e alla stampa) alcune righe o colonne.

Molti parametri importanti relativi all'impostazione del vostro lavoro possono essere modificati agendo nel riquadro Preferences.

Nel menu **Data** sono raccolte tutte le funzioni di manipolazione dati presenti in ogni foglio di calcolo che si rispetti. La procedura per lavorare su di un archivio è piuttosto semplice: si crea un range contenente i vari record ciascuno su di una linea, con i campi ben incolonnati; a questo punto sono possibili operazioni di ricerca (**Select Criteria** seguito da **Find**), di ordinamento (**Sort**) dei record o di un singolo campo (con **Sort Column**); si possono inoltre ottenere dati statistici sui contenuti del database (come il valore medio di un certo campo, oppure la deviazione standard) utilizzando le apposite funzioni.

AUTOMATISMI E PROGRAMMAZIONE

L'ultimo menu di questa schermata si occupa delle macro: con **Start Recording** tutte le vostre azioni saranno minuziosamente registrate fino a **Stop Recording**; quando avrete bisogno di ripetere la sequenze, dovrete soltanto richiamare la macro corrispondente con il comando **Play Back** e con il nome che le avete assegnato.

Il file requester è di tipo inedito: non è il più potente del mondo, ma risulta comunque sufficientemente pratico.

Se le macro non sono abbastanza potenti per voi, è sufficienti possedere l'interprete **ARexx** (oppure il WorkBench 2.0): potete associare uno script differente ad ogni cella del foglio, e lo script sarà eseguito (a meno che voi non diate istruzioni contrarie in **Preferences**) con un semplice doppio click. In questo modo, la programmabilità di «Professional Calc» supera tranquillamente quella di tutti i concorrenti in commercio.

Le funzioni con le quali potete mettere insieme le vostre formule sono di tutti i tipi: oltre che le tradizionali matematiche e trigonometriche, ce ne sono una dozzina per applicazioni finanziarie (calcolo di rate, interessi ed ammortamenti). Quelle statistiche lavorano prevalentemente su blocchi definiti come database; le alfanumeriche invece operano su celle contenenti del testo (**Left**, **Mid** e **Right** estraggono sottostringhe, **Upper** fa diventare maiuscole tutte le lettere componenti, **Trim** elimina gli spazi all'inizio).

Poiché le date sono memorizzate nell'inconsueto formato di cui si è detto, sono presenti funzioni come **Date** (che accetta come argomenti giorno, mese ed anno restituendo il numero corrispondente) e **Weekday** (che ritorna il codice del giorno della settimana associato alla data passata comr argomento); **Today()** è la data di oggi, e **Now()** l'ora dell'ultimo ricalcolo effettuato.

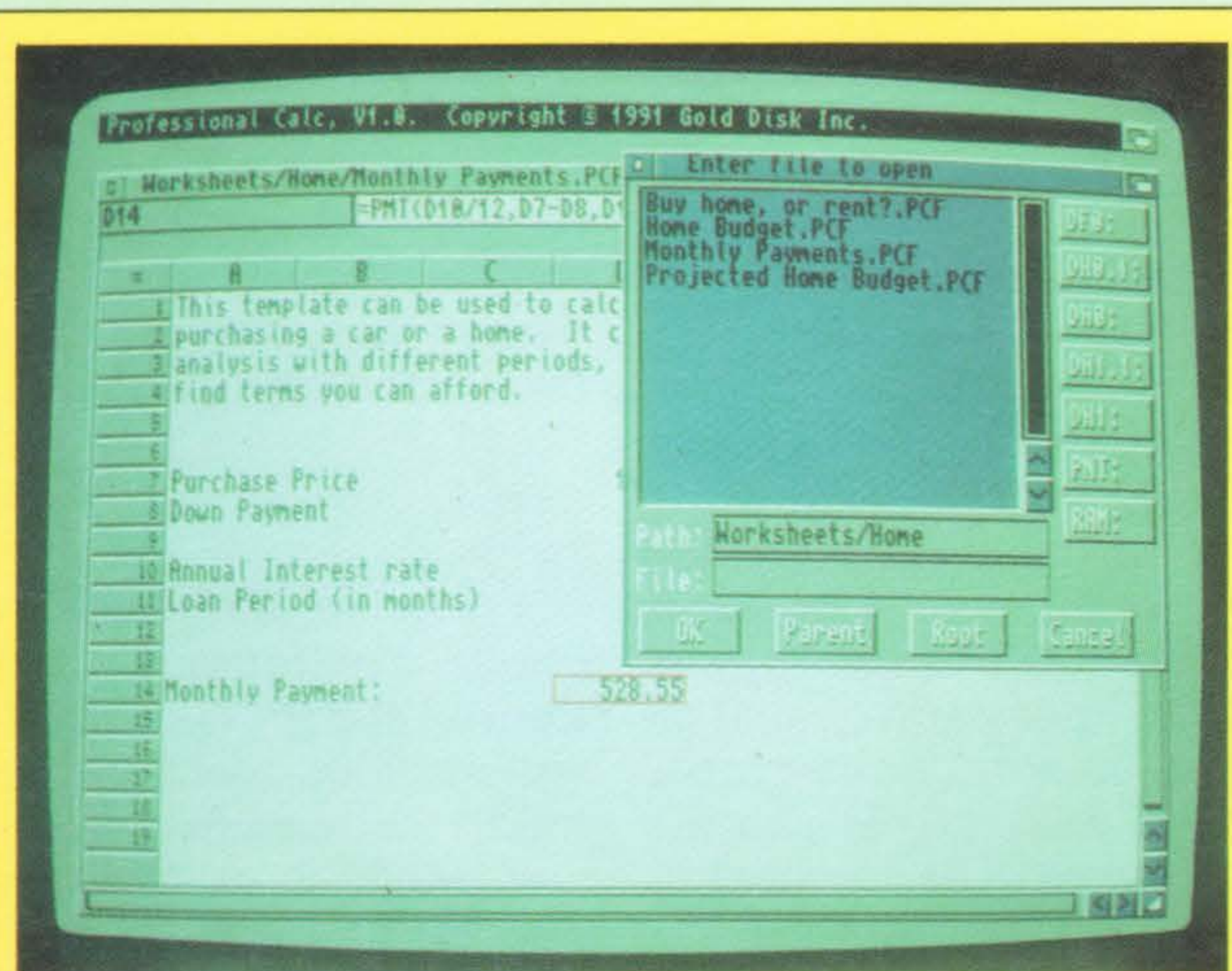
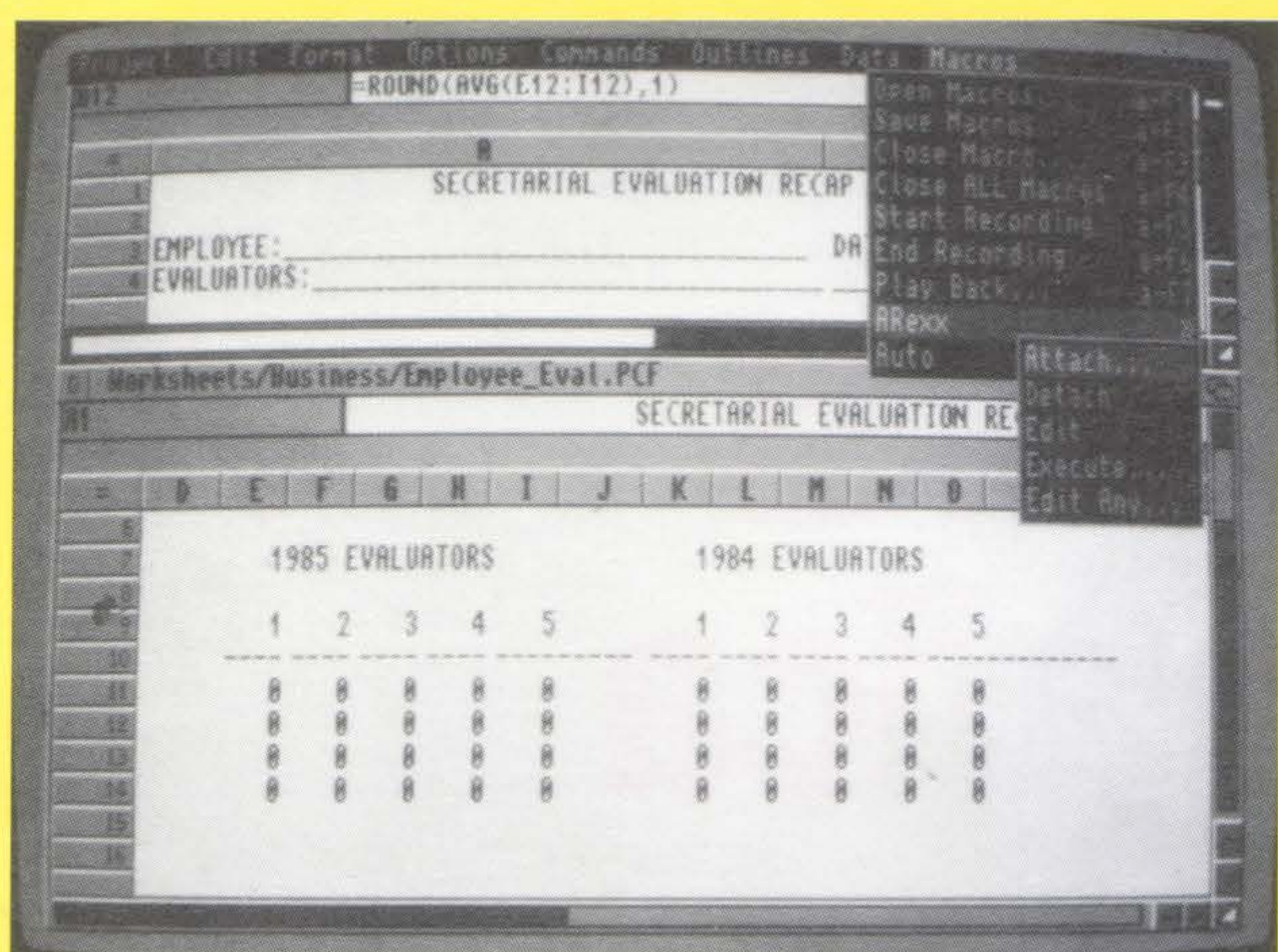
Le funzioni logiche hanno grande importanza nell'impostazione dei fogli di calcolo: **If(condizione, r1, r2)** vale **r1** se **condizione** è verificata, **r2** in caso contrario; **Isnumber**, **Iser**, **Isstring** ritornano vero o falso a seconda del tipo di dato contenuto nella cella passata come argomento: rispettivamente un valore numerico, un errore, una stringa.

Poiché la **If** prevede soltanto due casi (vero o falso), nel caso di controlli multipli sono necessari costrutti sintattici con molte If nidificate: se ad esempio vogliamo sapere che giorno della settimana è oggi, dovremo inserire in una cella (poniamo la A1) la formula **=WEEKDAY(TODAY())**, ed il nome desiderato verrà fuori da **=IF(A1=0,"Domenica", IF(A=1,"Lunedì", IF(A1=2,"Martedì", IF(A1=3,"Mercoledì", IF(A1=4, "Giovedì", IF(A1=5,"Venerdì","Sabato"))))))**.

Non mancano funzioni per modificare colore o stile del testo in una cella.

SE VI PIACE IL FAI DA TE

Questa dotazione è sicuramente sufficiente per la maggior parte delle esigenze; nulla vi vieta, però, di creare le vostre funzioni personali. A questo scopo basterà scrivere la formula corrispondente in una cella, ed assegnarle un nome con **New User Formula**, nel menu **Commands**. Unico inconveniente, la neonata funzione non accetta parametri; per aggirarlo, basterà inserirvi dei riferimenti relativi alle celle adiacenti. Ad esempio, non è possibile creare la funzione **RADDOPPIA(n)**, ma potete sempre definire una funzione che vale il doppio della cella soprastante. Volete vederla? Eccola: **=CELL(0,−1)*2**

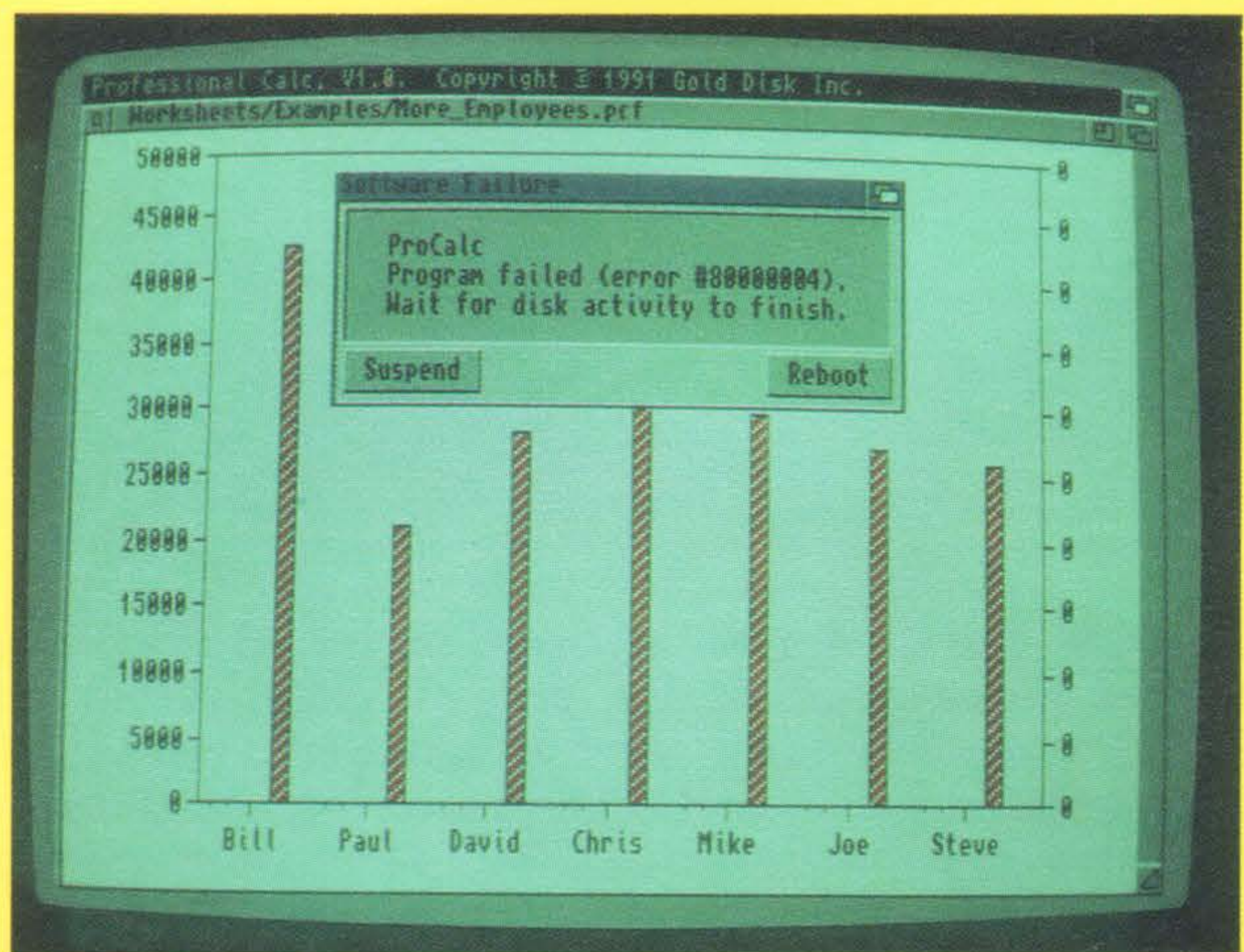
Per presentare i risultati che avete elaborato, «Professional Calc» vi mette a disposizione numerosi tipi di grafico: cartesiano, istogramma ed areogramma, con tutte le possibili variazioni sul tema, ivi comprese quelle tridimensionali. Per accedervi, assicuratevi di avere selezionato il range che vi interessa, e selezionate **New Graph** nel menu **Command**.

Le due finestre rappresentano inquadrature differenti dello stesso foglio, create con il comando View del menu Project.

Il grafico è personalizzabile fin negli aspetti più insignificanti: titoli, didascalie e legenda possono essere realizzati con font e stili differenti (compresi l'**Outline** e lo **Shadowed**); per gli assi possiamo stabilire il numero e/o la distanza tra le tacche di riferimento principali e quelle secondarie, nonché un massimo ed un minimo differenti da quelli proposti dal programma. Possiamo definire manualmente il colore di ogni singolo elemento (barra, linea o fetta di torta), gli eventuali retini da utilizzare per simulare la presenza di più colori sullo schermo, e così via; si possono eliminare i bordi della finestra (ottenendo così un grafico a tutto schermo, pronto per essere fotografato se ce ne occorre una diapositiva), oppure stampare l'immagine in una modalità a scelta fra tre (Postscript, Dot Matrix oppure **Enhanced Print**, che lavora sempre su stampanti ad aghi ma ha una resa migliore della modalità standard).

Dulcis in fundo, il grafico può essere salvato come immagine IFF, oppure nei formati «Aegis Draw» e naturalmente «Professional Draw». In particolare, questa ultima è la tecnica consigliata se occorre inserire dei grafici in documenti creati con un programma di DTP: segnaliamo che, oltre a «Professional Page», anche «Pagestream» può importare i file in questione.

IL ROVESCIO DELLA MEDAGLIA

Dopo tante lodi, è venuto il momento di parlare dei lati oscuri di questo software; il più inquietante è il fatto che, con una directory FONTS: comprendente più di una cinquantina di tipi di carattere, è sufficiente premere Amiga-F per causare il blocco del sistema; considerati i risultati della nostra prova, però, ci sentiamo di affermare che si tratta dell'unico vero bug riscontrato in «Professional Calc».

I due riquadri per i parametri di stampa: PostScript a sinistra, Dot Matrix a destra.

Nel caso il device logico FONTS: sia piuttosto affollato, siate pronti ad uno spettacolo simile a questo...

Gli altri appunti che si possono muovere sono più che altro di ordine estetico-pratico: ad esempio, i numeri delle righe vengono visualizzati con un font davvero piccolissimo (cinque punti di altezza), la cui lettura risulta molto ostica in interlacciato e comunque stancante in caso contrario. Sarebbe bastato utilizzare lo stesso font delle celle, così come fa «Excel».

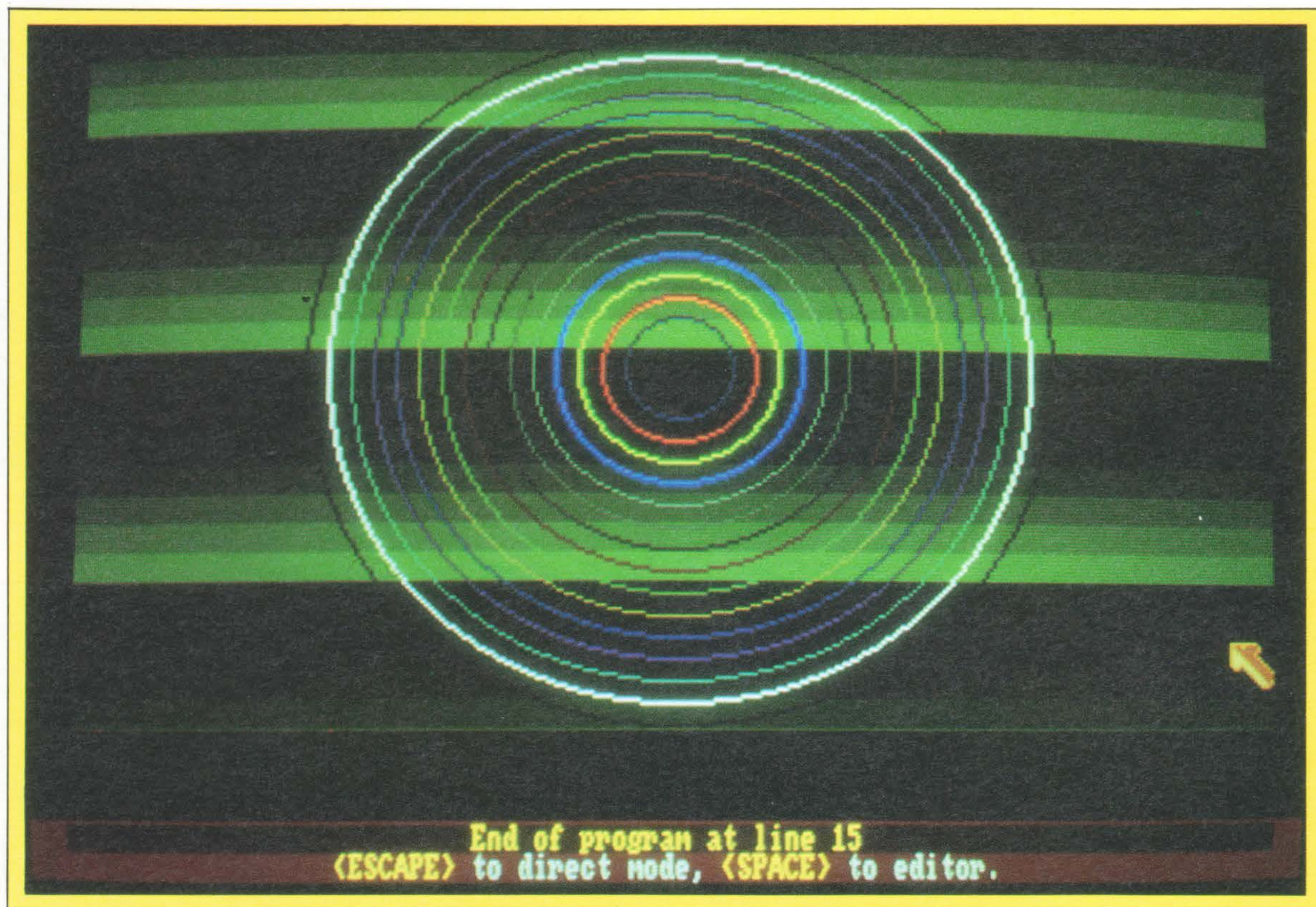
Potremmo continuare: per cambiare il formato di schermo occorre utilizzare un'opzione che compare soltanto quando nessun foglio o menu grafico è aperto; se dunque desideriamo passare in interlace, ci tocca salvare il nostro lavoro, chiuderlo, cambiare modalità grafica, e ricaricarlo.

Anche la maggior parte delle funzioni riguardanti le date risultano purtroppo di scarsa utilità a causa dell'inglese: passi per un programma in lingua straniera, ma almeno le date che compaiono nei nostri documenti dovrebbero essere modificabili! La Commodore prevede il supporto di lingue diverse dall'inglese nella prossima versione del sistema operativo: ci auguriamo che, con questo, problemi del genere siano destinati a scomparire.

Precisiamo, comunque, che si tratta di inconvenienti di lieve entità, che non inficiano il giudizio più che positivo maturato circa questo prodotto della Gold Disk, destinato ad affermarsi come standard e (ci auguriamo) a stimolare la concorrenza nella produzione di spreadsheet sempre più potenti e capaci di sfruttare appieno le peculiarità di Amiga. □

Primi passi con la grafica

Prosegue il viaggio nel mondo di Amos con lo studio delle funzioni grafiche più semplici che l'interprete mette a disposizione.

di PAOLO BOZZO
Seconda parte

L'interprete Amos offre molte possibilità grafiche, eventualmente potenziabili mediante la recente estensione apportata al linguaggio, «**Amos 3D**»: non è un caso che uno dei principali campi di applicazione di Amos sia quello dei giochi, nei quali la grafica riveste un ruolo fondamentale.

Come abbiamo fatto osservare la volta scorsa, esiste in Amos una incompatibilità di fondo tra l'istruzione **PRINT** (o le altre ad essa associate e già descritte) e le istruzioni di tipo grafico, tanto da farci parlare di un «modo testo» e di un «modo grafico». In pratica dobbiamo pensare all'esistenza di due cursori: quello usato per PRINT e quello grafico per tracciare linee, cerchi, etc. (ma anche testo); con **LOCATE** posso piazzare il cursore di print in un punto e con **GR LOCATE** mettere contemporaneamente il cursore grafico altrove:

```
Locate 10,10
Print "Cursore TESTO"
Gr Locate 10,10
Text ,, "Cursore GRAFICO"
```

Se fate girare questo listatino, vi accorgerete che le due scritte compaiono in due punti diversi dello schermo, perché il primo LOCATE misura i caratteri ed il secondo i pixel.

È così possibile riscrivere in «modo grafico» anche l'esempio di «saluti colorati» pubblicato la volta scorsa:

```
Curs Off
Flash Off
For I=0 To 15
   Ink I, I+1
   P=(I+1)*8
   Text P,P, "Hello, Amos Basic!"
Next
```

Come potete ben osservare, a parte alcuni aspetti di «contorno», le istruzioni che abbiamo utilizzato sono completamente diverse da quelle della volta scorsa. Si noti bene che, essendo di tipo grafico, queste istruzioni per cambiare colore o posizionare il cursore non agiscono solo sul comando **TEXT** (usato per scrivere una stringa), ma anche sul tracciamento di linee, cerchi, e così via.

Il comando TEXT per scrivere un testo in modo grafico è semplicissimo, richiede soltanto la posizione delle coordinate del cursore X ed Y (se abbiamo già usato GR LOCATE, l'operazione può essere evitata) e la stringa da stampare sullo schermo. Contrariamente a quanto fa PRINT, TEXT non manda a capo il cursore una volta che ha stampato una stringa. Dobbiamo pensarci noi fornendo delle nuove coordinate: ricordatevi che vengono misurate in pixel e che quindi vanno incrementate di 8 (e non di uno) per spostarle di un carattere in ogni direzione.

L'istruzione **INK** (che svolge all'incirca la stessa funzione dell'istruzione COLOR di AmigaBasic) serve invece per cambiare i colori utilizzati dal cursore grafico: primo piano, sfondo e contorno (noi abbiamo specificato solo primo piano e sfondo: in questo senso Amos consente, come AmigaBasic, di trascurare alcuni parametri per lasciarli al valore di *default*).

Amiga può stampare testo con diversi *font* (ci dobbiamo adeguare alla pessima abitudine di usare termini inglesi; *font*, cioè «fonte», sta ad indicare una serie particolare di caratteri alfabetici, come per esempio i caratteri «gotici»). Questi font, inoltre, possono essere mostrati in stile normale, *italic* (corsivo), sottolineato, **bold** (grassetto) oppure in qualsivoglia combinazione di questi stili. Sono operazioni possibili con AmigaBasic solo a prezzo di assurdi contorsionismi, relativamente semplici invece con Amos.

Se, per esempio, all'inizio dell'ultimo esempio aggiungiamo:

```
Set Text 4
```

vedremo le scritte comparire in corsivo. Il parametro passato a **SET TEXT** dev'essere una maschera di bit, così:

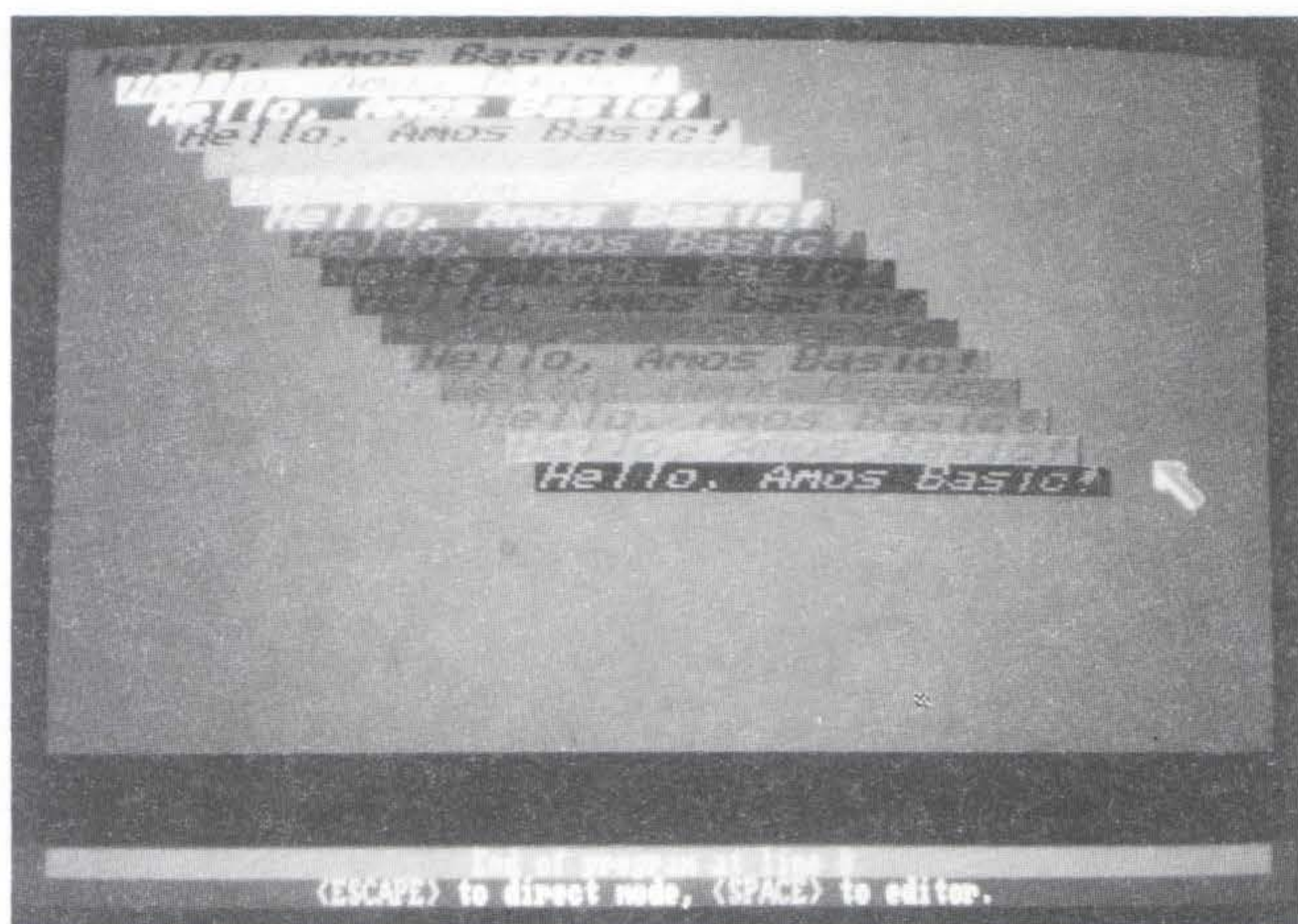
BINARIO	DECIMALE	EFFETTO
%0000	0	normale
%0001	1	sottolineato
%0010	2	grassetto
%0100	4	corsivo

Ricordiamo che in Amos il segno % indica un valore binario.

In questo modo sarà possibile ottenere anche combinazioni, come %0011 (3 decimale), che darà uno sottolineato e grassetto insieme.

GESTIONE DEI FONT

Per quanto riguarda i font è invece possibile avere una lista di tutti quelli disponibili chiamando la funzione **GET FONTS** (ci si può anche limitare a quelli su Rom con **GET ROM FONTS**, o a quelli su disco con **GET DISC FONTS**); come risultato potremo consultare l'array **FONT$ ()** che conterrà dati relativi ad ogni font. Per sapere quali font sono presenti su di un disco, fate girare questo listato:

```
Get Fonts
I=1
While Font$(I)<>""
   Print Font$(I)
   Inc I          ' (è come scrivere I=I+1)
Wend
```

Le stringhe di questo array hanno un formato standard: negli spazi a partire dal primo ci sarà il nome del font di caratteri; in quelli a partire dal trentesimo, l'altezza del font; infine, a partire dal trentaquattresimo avremo la parola ROM o DISC ad indicare che il font si trova su ROM o su disco.

Quando si vuole caricare un font da disco bisogna usare l'istruzione **SET FONT** seguita da un numero che indica a quale valore corrisponde l'indice dell'elemento che contiene la definizione del font che vogliamo caricare; per esempio, se vogliamo caricare il «ruby.font» con altezza di quindici pixel e il 28° elemento dell'array, FONT$(28), si presenta così:

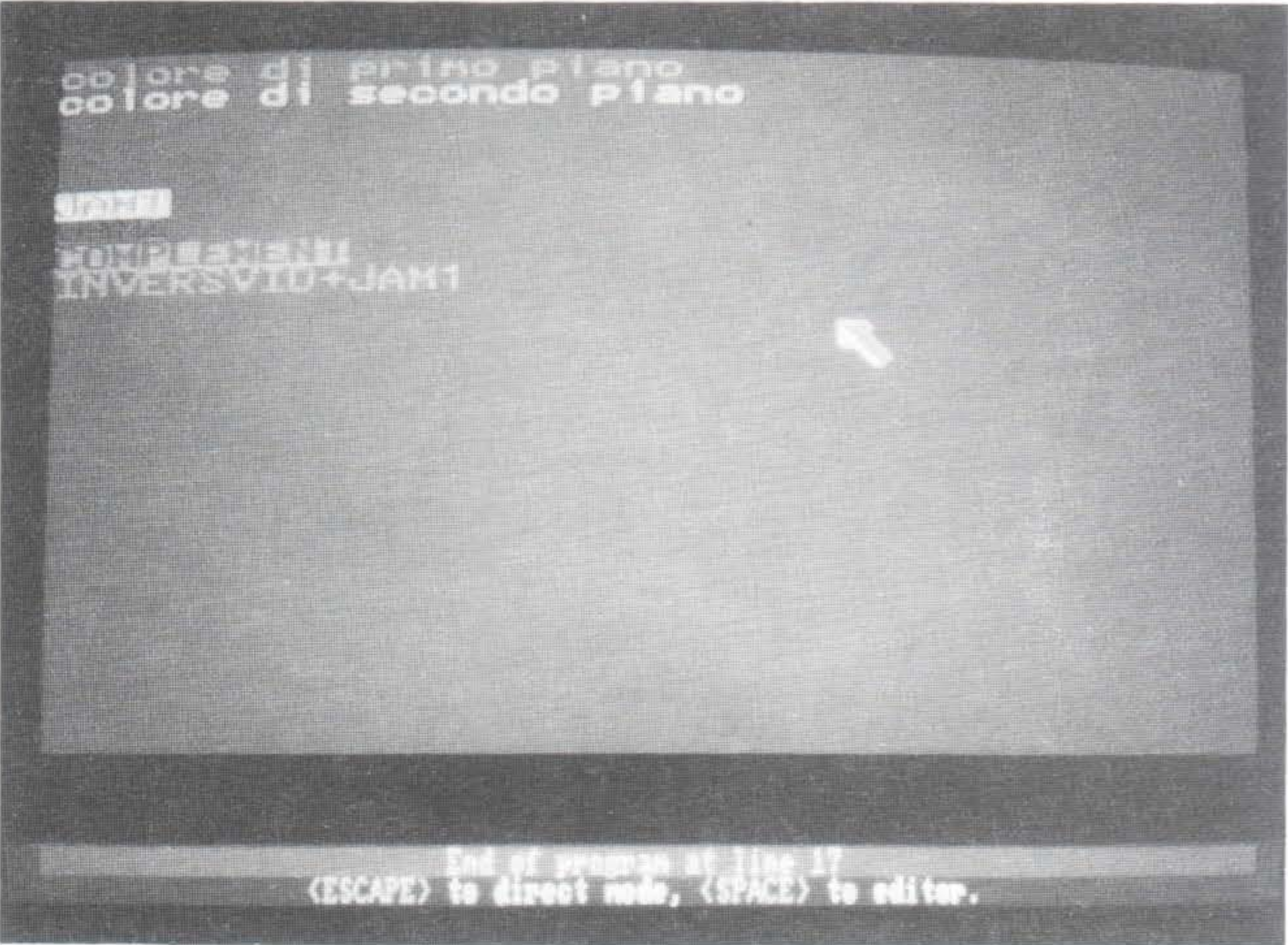
"Ruby.font   15 DISC"

dovremo quindi usare l'istruzione SET FONT 28.

Per quanto il sistema funzioni correttamente, non ci sembra affatto pratico, soprattutto perché non si può sapere in precedenza quale sarà il «numero magico» e si è quindi costretti a chiamare GET FONTS, con una considerevole perdita di tempo (pensate se avessimo 50 font sul disco!). Occorrerà inoltre analizzare tutte le stringhe

dell'array FONT$ per trovare quella giusta. Ammettiamo di dover caricare il «Ruby.Font» di cui sopra. Il procedimento sarà questo:

```
Get Disc Fonts
I=1
SEARCH_FONT$="RUBY.FONT              15 DISC"
Repeat
   If SEARCH_FONT$=Upper$(Font$(I)) Then TROVATO=-1
   Inc I
Until Font$(I)="" or TROVATO
If Not TROVATO
   Print "non trovato font Ruby"
Else
   Cls : Set Font I-1
   Text 0,15,"Questo è Ruby!"
End If
```

Piuttosto laborioso, come si vede; e le cose si complicheranno se vorremo adattare la routine a qualsiasi font (consigliamo al lettore questo esercizio). Notate il ciclo REPEAT ... UNTIL che termina quando l'espressione dopo UNTIL è vera.

LINEE E FIGURE GEOMETRICHE

Come è classico di ogni linguaggio BASIC, anche Amos prevede una serie di istruzioni grafiche per tracciare linee e figure geometriche sullo schermo. Ai neofiti diciamo subito di non pensare a tracciare immagini complesse con questi sistemi, che in questi casi non garantiscono sempre la sufficiente velocità, la versatilità e la praticità; tuttavia questi comandi sono utili in casi particolari, ad esempio per creare una speciale interfaccia-utente o dei semplici disegni di tipo geometrico. Abbiamo elencato i comandi principali in un quadro separato.

Piuttosto facile risulta anche fissare una maschera per il tracciamento di queste figure: basta usare **SET LINE** seguito da una maschera di 16 bit (un numero intero decimale). Per esempio, dopo la seguente istruzione:

```
SET Line %1100110011001100
```

tutte le linee appariranno tratteggiate. Per tornare allo «stato normale» basterà SET LINE −1. Il limite di SET

```
Flash Off : Curs Off
Dim GMOD$(3)
GMOD$(0)="JAM1"
GMOD$(1)="JAM2"
GMOD$(2)="COMPLEMENT"
GMOD$(3)="INVERSVID+JAM1"
PRIMO_PIANO=5
SECONDO_PIANO=6
Ink PRIMO_PIANO
Text 0,8, "colore di primo piano"
Ink SECONDO_PIANO
Text 0,16,"colore di secondo piano"
Ink PRIMO_PIANO, SECONDO_PIANO
For I=0 To 3
   Gr Writing 2^I
   Text 0,48+(I*8), GMOD$ (I)
Next
```

LINE consiste nel fatto che la maschera è lineare; il comando **SET PATTERN** invece permette di definire una maschera a due dimensioni, anche presa da uno sprite (ne riparleremo in seguito, quando tratteremo gli sprite).

OCCUPIAMOCI DEI COLORI

All'istruzione INK per cambiare i colori di tracciamento se ne associano altre per definire o modificare totalmente o parzialmente la tavolozza (*palette* in inglese), o per selezionare il modo di tracciamento. INK consente di definire il colore delle tre penne di tracciamento (primo piano, secondo piano, contorni); ma la libreria grafica di Amiga prevede quattro modi di tracciamento, impostabili direttamente con la funzione *SetDrMd()* (della graphics.library, non di Amos!), I modi sono: **JAM1**, che fa sì che venga utilizzato solo il colore di primo piano; **JAM2**, che fa sì che vengano utilizzati il colore di primo piano e quello di secondo piano; **COMPLEMENT** (in Amos chiamato **XOR**), che effettua una operazione di complemento sui bit della bitmap destinazione; **INVERSVID**, che seleziona il modo «inverso» e va usato insieme a JAM1 o a JAM2.

Amos permette di selezionare il modo di tracciamento grazie all'istruzione **GR WRITING** seguita da una maschera di bit; ecco i bit da utilizzare:

BINARIO	DECIMALE	EFFETTO
%0000	0	JAM1
%0001	1	JAM2
%0010	2	XOR
%0100	4	INVERSVID

Se questo sistema vi ha causato qualche perplessità, non spaventatevi; consultate il nostro listatino «Modi.amos» (presente, come sempre, anche sul dischetto allegato alla rivista) e fate esperimenti con esso per ottenere tutte le possibili combinazioni di bit e di colori (la maschera di bit non deve comunque andare oltre %111 binario = 7 decimale).

Per fissare i colori della nostra tavolozza abbiamo le istruzioni **COLOUR** e **PALETTE**. COLOUR (che corrisponde approssimativamente al comando PALETTE di AmigaBasic, tanto per complicare le cose a chi era abituato all'interprete standard) modifica l'impostazione di un singolo colore della tavolozza.

Come è noto, i colori di Amiga sono indicati da tre valori (RGB = rosso, verde, blu) a 16 bit, riconducibili ad un numero esadecimale a tre cifre (in Amos i numeri esadecimali sono preceduti dal segno $, esattamente come in assembler); ognuno di questi tre valori indicherà la quantità di rosso, di verde o di blu da utilizzare; in totale saranno possibili 4096 colori, che andranno da $000 (nero) a $FFF (bianco). Ecco un esempio:

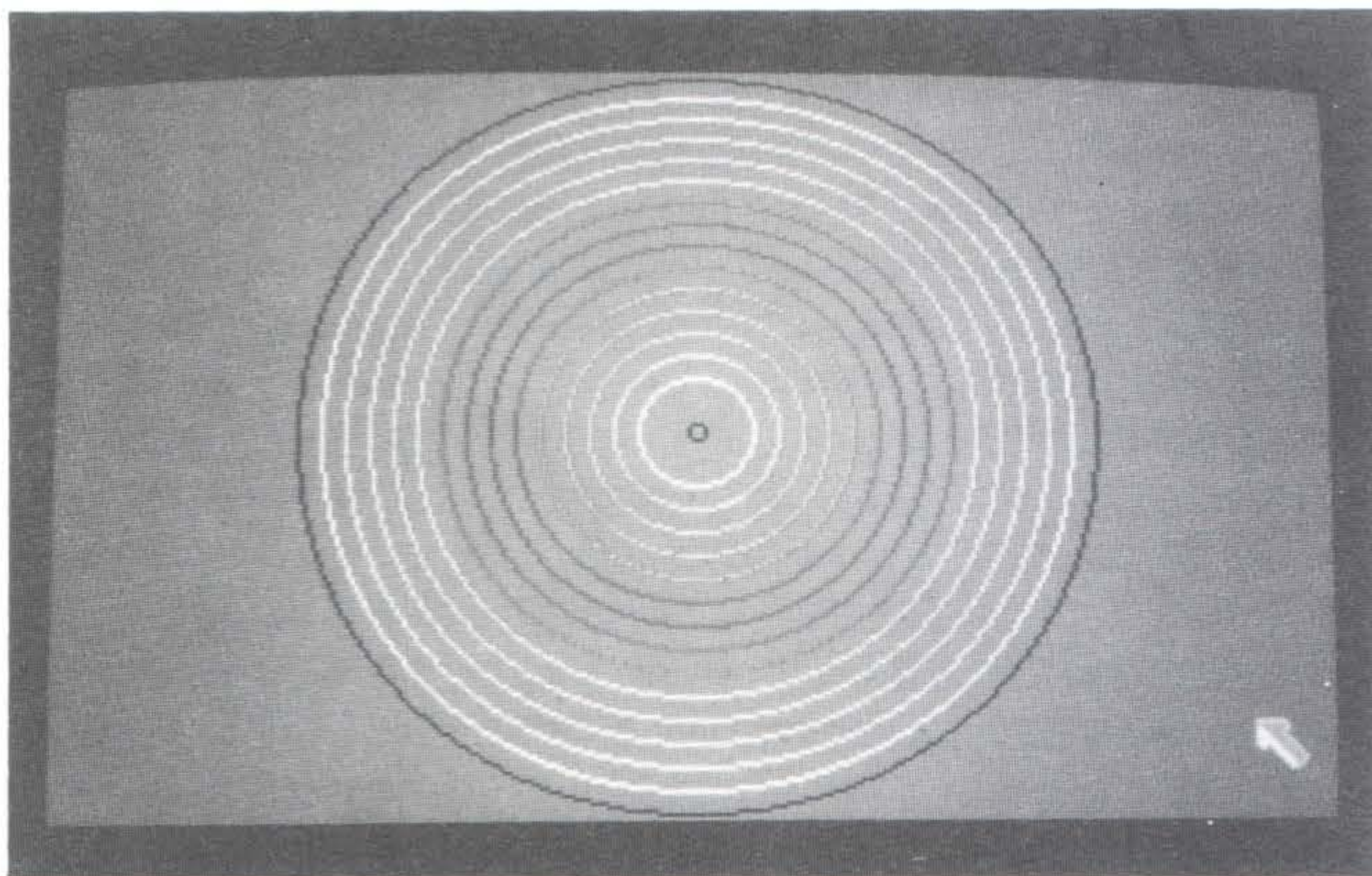
```
Colour 5, $9AB
```

Con questa linea diremo che il colore n. 5 della tavolozza deve avere 9 parti di rosso, 10 (esadec. = A) di verde e 11 (esadec. = B) di blu: in questo caso il colore risultante sarà un grigio chiaro tipo carta-da-zucchero.

L'istruzione **PALETTE** (che non ha corrispondente in AmigaBasic) consente di evitare una lunga sequela di istruzioni COLOUR, consentendoci di modificare l'intera tavolozza o parte di essa (i valori saltati non vengono modificati). Anche in questo caso, un esempio può essere più chiaro di tante spiegazioni. Con la linea:

```
Palette ,,$000,,$FFF
```

si mette il colore n. 2 a nero ed il colore n. 4 a bianco (contate le virgole e tenete presente che i colori cominciano da 0).

Le chance offerte dai singoli colori non si fermano qui. Avete presente il *color cycling* di «Deluxe Paint»? La possibilità di far ruotare fra di loro i colori, creando così anche semplici animazioni, è offerta da Amos con il comando **SHIFT** seguito da **UP** se devono ruotare verso l'alto, o da **DOWN** se devono ruotare verso il basso. SHIFT UP (o DOWN) prevede come parametri il ritardo (in cinquantesimi di secondo — tra una sostituzione e l'altra), il colore di inizio, il colore di fine ciclaggio ed un *flag* (1 o 0) che controlla il tipo di ciclaggio (consigliamo il valore 1). Per far smettere il ciclaggio basta **SHIFT OFF**. Se avete da creare effetti speciali come pioggia, o fiamme, oppure onde, cercate di realizzarli con «Deluxe Paint» ed il color-cycling, e poi riproduceteli con il comando SHIFT.

Con Amos si possono ottenere effetti grafici notevoli con poche righe di programma.

Caricare un disegno creato con «Deluxe Paint» è per Amos, come vedremo, un gioco da ragazzi.

```
Flash Off : Curs Off
For I=1 To 99 Step 6
   If I>0 Then Ink I/6
   Circle 160, 100, I+1
Next
Shift Up 2,2,15,1
Wait Key
Shift Off
Fade 10
End
```

CREARE UN ARCOBALENO

Un altro effetto scenografico non male consiste nell'utilizzare l'effetto di dissolvenza con il comando **FADE**. FADE può essere seguito da un singolo parametro indicante il tempo (in cinquantesimi di secondo) che dev'essere impiegato per far sfumare tutti i colori. In alternativa si può ottenere una «dissolvenza incrociata», che sfuma da uno schermo all'altro, aggiungendo al primo parametro l'espressione TO S, dove S è il numero dello schermo che va visualizzato dopo la dissolvenza.

Continuiamo la lista di questi «effetti speciali», ricordando che è possibile creare un «effetto arcobaleno»; si tratta di una opportunità offerta da Amiga, che troviamo qualche volta su programmi scritti in assembler e che è a portata di mano con Amos, sia pure con qualche complicazione. I comandi da utilizzare sono: **SET RAINBOW** (con cui definiamo le caratteristiche dell'«arcobaleno») e **RAINBOW** (con cui creiamo l'effetto). A questi due comandi, indispensabili, si aggiunge **RAIN** (che preleva/modifica una singola linea dell'arcobaleno). Un esempio:

```
Set Rainbow 0, 1, 64, "(8, 1, 12)", "",""
   Rainbow 0, 15, 1, 255
```

Per entrambi i comandi il primo parametro indica il numero del «rainbow». Il secondo parametro di SET RAINBOW indica il numero del colore sul quale si vuol creare l'effetto; il terzo parametro è la dimensione della tabella in cui salvare i colori-arcobaleno, ed è un numero che varia da 16 a 54400. Solitamente questo valore è inferiore all'altezza del «rainbow» in modo che l'effetto si ripeta più volte sullo schermo; il quarto, il quinto ed il sesto parametro di SET RAINBOW sono delle stringhe che si riferiscono rispettivamente ai colori rosso, verde e blu; il formato di queste stringhe deve essere:

(n1, n2, n3)

dove n1 è l'altezza in pixel di ciascuna «striscia di colore» dell'effetto «arcobaleno»; n2 è la differenza di gradazione tra un colore e l'altro (consigliamo di tenere basso questo valore, in modo da ottenere effetti molto «sfumati» e più piacevoli); l'ultimo numero (n3) indica il numero di volte in cui va ripetuta l'operazione.

Il secondo parametro di RAINBOW indica uno sco-

ALCUNI COMANDI GRAFICI

DRAW
sintassi: DRAW minx, miny TO maxx, maxy
funzione: tracciare una linea da minx, miny a maxx, maxy (se minx, miny non specificati, parte dal punto in cui si trova il cursore grafico).

BOX
sintassi: BOX minx, miny TO maxx, maxy
funzione: tracciare un rettangolo vuoto da minx, miny a maxx, maxy.

BAR
sintassi: BAR minx, miny TO maxx, maxy
funzione: tracciare un rettangolo pieno da minx, miny a maxx, maxy.

POLYLINE
sintassi: POLYLINE x1, y1 TO x2, y2 TO x3, y3 ...
funzione: tracciare un poligono vuoto da x1, y1 a x2, y2 ... e via di seguito.

POLYGON
sintassi: POLYGON x1, y1 TO x2, y2 To x3, y3 ...
funzione: tracciare un poligono pieno da x1, y1 a x2, y2 ... e via di seguito.

CIRCLE
sintassi: CIRCLE x, y, r
funzione: tracciare un cerchio con centro in x, y e raggio di dimensioni r.

ELLIPSE
sintassi: ELLIPSE x, y, r1, r2
funzione: tracciare una ellisse con centro in x, y e raggi di dimensioni r1 ed r2.

PAINT
sintassi: PAINT x, y, modo
funzione: riempire del colore di primo piano un'area a partire dal punto x, y; se modo=1 il colore si arresta se incontra pixel di colore diverso dal proprio. Si veda anche il comando INK.

stamento relativo al primo colore della tabella; per ottenere un effetto di «arcobaleno mobile» basta cambiare questo colore. Gli ultimi due parametri di RAINBOW indicano in coordinate *hardware* l'inizio e la fine della zona verticale dello schermo occupata dall'effetto (1 e 255 dell'esempio valgono per uno schermo PAL non interlacciato).

PER APRIRE GLI SCHERMI

Le operazioni che abbiamo svolto finora nei nostri esempi utilizzano lo schermo di *default* di Amos, uno schermo a bassa risoluzione di 320x200 pixel che viene aperto automaticamente dall'interprete (o dal programma compilato, se usate il compilatore). A chi non fosse soddisfatto di questo schermo o ne volesse aprire altri, Amos offre sia la possibilità di alterare lo schermo di *default*, sia quella di aprirne altri, fino a sette.

Invitiamo comunque i nuovi adepti della programmazione a non abusare di queste caratteristiche, vale a dire a non aprire troppi schermi con molti colori: come su ogni computer grafico, anche su Amiga gli schermi occupano una notevole quantità di memoria che, tra l'altro, dipende dalla risoluzione, dal numero di colori, dal modo grafico utilizzato.

Amos non ricorre ad Intuition per creare gli schermi e, così, probabilmente risparmia un po' di memoria, oltre che ottenere maggior flessibilità; ma la grafica comunque avrà bisogno di molta memoria Chip. Aggiungiamo che anche se siamo tanto fortunati da avere sulla nostra macchina un mega o due di memoria Chip, se intendiamo diffondere il nostro *software* dovremo tener presente che ci sono in circolazione ancora molti 500 con 512K di Chipmemory. Ritorneremo comunque presto sul problema della gestione della macchina con Amos.

Per aprire uno schermo nuovo basta l'istruzione **SCREEN OPEN** seguita da cinque parametri indicanti il numero dello schermo (quello di default è 0, gli altri vanno da 1 ad 8), la larghezza, l'altezza, il numero di colori ed il modo grafico. Particolarmente semplice sarà anche chiudere uno schermo con **SCREEN CLOSE**, seguito dal numero dello schermo. Per esempio, se vogliamo che lo schermo di *default* sia simile a quello del WorkBench (alta risoluzione non interlacciato), basterà:

SCREEN OPEN 0,640,512,4,HIRES

Occorre solo fare un po' di attenzione all'ultimo parametro che potrà essere: **HIRES** (alta risoluzione, larghezza solitamente 640, non più di 16 colori); **LORES** (bassa risoluzione, larghezza solitamente 320, non più di 32 colori); **LACED** (interlacciato); **EHB** (*Extra-Half-Bright:* colori raddoppiati da 32 a 64); **HAM** (il famoso modo *Hold and modify* che permette 4096 colori).

I modi EHB ed HAM sono piuttosto attraenti da usare su Amiga, anche se bisogna fare i conti con le necessità di memoria (circa 48K con uno schermo 640x200). Il modo EHB permette appunto 64 colori con sei piani di bit: l'unica limitazione consiste nel fatto che i colori dal 32 al 63 sono una copia a luminosità dimezzata dei colori dallo 0 al 31; questa caratteristica può essere una limitazione tale da farci adottare il modo LORES a 32 colori: solo in qualche caso, quando servono particolari effetti di luci ed ombre, potrebbe tornare utile.

Estremamente potente ed affascinante è il modo HAM, che permette di utilizzare tutti i 4096 colori di Amiga contemporaneamente sullo stesso schermo; questo modo è raccomandabile nella riproduzione di fotografie, che acquistano così un realismo impressionante, o per scopi particolari; molti programmi di disegno (ed adesso anche «Deluxe Paint IV») permettono il modo HAM, dunque si potrà realizzare con essi la nostra schermata HAM e poi caricarla con Amos.

Le limitazioni del modo HAM sono poche: per esempio, una certa incompatibilità con il comando FADE e con lo *scrolling* orizzontale.

Si può caricare una schermata IFF con una semplicissima istruzione **LOAD IFF**, seguita dal nome del file; questa istruzione ha un secondo parametro opzionale che indica il numero dello schermo (se lo schermo con tale numero non è stato ancora aperto, verrà aperto esattamente come se si fosse chiamato OPEN SCREEN). Si può anche salvare una schermata con l'istruzione **SAVE IFF**, seguita dal nome del file, che salverà la schermata corrente in modo compresso salvando anche tutte le informazioni eventualmente utili ad un successivo riutilizzo con Amos.

DILEMMA TELEMATICO

Sarei interessato all'acquisto di un modem, ma purtroppo non sono abbastanza informato in questo campo.

Che differenza c'è tra un modem che trasmette a 1200 baud e uno che trasmette a 2400 baud? Un modem a 2400 baud può trasmettere anche a 1200?

Per quanto riguarda le spese, è necessario pagare una tassa alla Sip? A quanto ammontano i costi per l'uso di un modem?

Andrea Salerno - Trapani

Il termine «baud» rappresenta l'unità di misura della velocità di trasmissione in campo telematico, ed equivale al numero di bit che il modem può trasmettere al secondo. Un modem a 2400 baud può perciò trasmettere 2400 bit al secondo tramite la linea telefonica; un modem a 1200 opera quindi a metà della velocità di un modem a 2400.

La maggior parte dei modem in circolazione, salvo eccezioni rarissime, è compatibile verso il basso: ovvero, è in grado di trasmettere a velocità inferiori a quella massima. Un modem a 2400 baud quindi può funzionare anche a 1200, anche se ovviamente è quasi sempre preferibile operare alla massima velocità possibile.

I costi di utilizzo di un modem sono limitati a quelli della bolletta telefonica: la spesa di un collegamento di 10 minuti è quindi quella degli scatti (urbani o interurbani) conteggiati per quella durata.

Fino a qualche mese fa, l'utilizzo di apparecchiature per la trasmissione di dati sulla linea telefonica presupponeva inoltre il pagamento di una tassa di concessione governativa (nell'ordine di circa 200.000 lire annue). Questo balzello è però ora stato abolito.

Occorre tenere presente comunque che, almeno teoricamente, gli unici apparecchi che possono essere collegati alla linea telefonica sono quelli dotati dell'apposita omologazione SIP.

EMULATORI E 500 PLUS

Possiedo un Amiga 500 Plus, con Kickstart 2.0, e vorrei acquistare una scheda o un programma di emulazione MsDos. Non sono però sicuro che le schede in commercio siano compatibili con il mio computer: alcuni negozianti ai quali mi sono rivolto mi hanno riferito che molte periferiche per l'Amiga 500 normale non funzionano sull'Amiga 500 Plus.

Marcello Algani - Ravenna

L'hardware di Amiga 500 Plus è effettivamente piuttosto diverso da quello del suo predecessore, ed alcune schede hardware (emulatori, espansioni...) presentano problemi di compatibilità. I maggiori produttori di hardware per Amiga sono al corrente dell'inconveniente e sono già corsi ai ripari, mettendo in circolazione nuove versioni dei loro accessori specifiche per il 500 Plus.

Un caso eclatante è la scheda Amiga Action Replay, che sul 500 Plus non funziona e per la quale la Datel ha annunciato una versione corretta.

Nel settore degli emulatori esistono già nuove revisioni sia della Power Pc Board che della AtOnce. La casa produttrice di quest'ultima, la tedesca Vortex, ha inoltre annunciato che distribuirà gratuitamente un upgrade, per renderla compatibile con il KickStart 2.0.

QUALCHE DUBBIO SUL CDTV

Dopo aver letto il vostro articolo sul CDTV apparso nel numero 34 mi ero messo il cuore in pace, convincendomi che si trattava di un modello autonomo e non di una periferica che avrei potuto collegare al mio Amiga 2000. Ma, a distanza di qualche settimana, ho avuto modo di sfogliare una specie di minivocabolario allegato ad un'altra rivista Amiga uscita dopo la vostra, nel quale alla voce CDTV ho trovato scritto che si tratta di «una periferica SCSI attualmente disponibile in due versioni: una interna per Amiga 2000 e derivati, ed una esterna per Amiga 500 e 1000».

A questo punto non capisco più nulla e chiedo una spiegazione: potrò mai usare i programmi per CDTV anche con il mio Amiga 2000 o devo rassegnarmi definitivamente?

Vittorio Lamberti - Modena

Non sappiamo il motivo per cui sia stata usata in quel caso una simile definizione per descrivere il CDTV, ma una visita ad un qualsiasi rivenditore Commodore può fugare definitivamente i tuoi dubbi, permettendoti di verificare con i tuoi occhi che si tratta di un prodotto del tutto indipendente e non di una periferica per modelli Amiga già esistenti.

Non c'è ragione comunque di disperare: anche sul tuo Amiga sarà tra breve possibile sfruttare il software su Cd-Rom per il CDTV. Commodore ha presentato un modello di lettore CD-Rom per Amiga (chiamato A690). Gli esemplari intravisti per il momento sono in versione esterna per Amiga 500, non dovrebbe tardare ad apparire una versione interna anche per Amiga 2000 e 3000.

PoNGo 1.0: trasformazioni 3D

Un nuovo tool di modellazione solida che consente di realizzare sorprendenti animazioni ed inediti effetti con «Imagine» e con molti altri pacchetti di grafica 3D.

di LORENZO ORLANDINI

Il più popolare programma di modellazione e visualizzazione solida disponibile su Amiga, e secondo molti anche il più potente, è senza dubbio «**Imagine**» della Impulse, al quale AmigaByte ha già dedicato moltissimo spazio nei fascicoli precedenti.

Giunto recentemente alla versione 2.0, questo pacchetto software offre caratteristiche di tutto rispetto sia per quanto riguarda l'editing dei solidi, che per il rendering (Ray-Tracing). Le uniche limitazioni evidenti del modellatore risiedono nell'estrema difficoltà di gestire complesse trasformazioni dinamiche, ovvero di far cambiare forma ad un oggetto in maniera realistica, all'interno di una sequenza animata.

Per far fronte a questa lacuna di «Imagine», comune anche alla maggior parte degli altri software di ray tracing, è stato creato «**PoNGo 1.0**», un programma di modellazione solida concepito per effettuare le più svariate trasformazioni (torsioni, deformazioni, metamorfosi, etc.) sugli oggetti nei formati supportati da «Imagine» e dal suo predecessore «**Turbo Silver 3.0**» (sempre della Impulse).

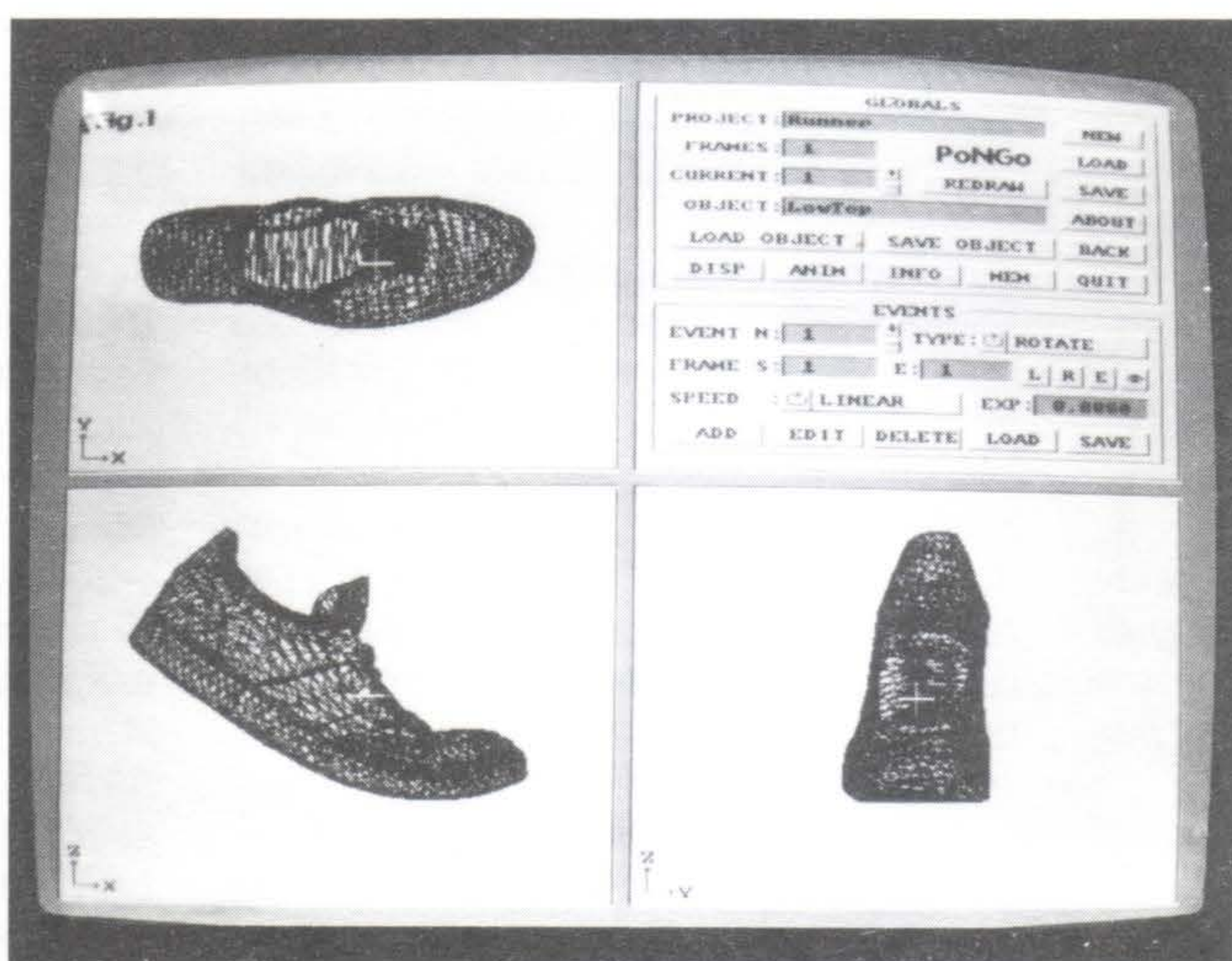



Fig. 1: l'ambiente di lavoro di «PoNGo»: tre finestre mostrano l'oggetto in memoria da tre punti di vista.

LA COMPATIBILITÀ

Il formato degli oggetti generati da «Pongo» è quello binario TDDD, ed il programma supporta tutte le versioni di «Imagine», compresa la recente 2.0: ricorrendo inoltre ad utility come «**InterChange**» della Syndesis o «**Pixel3D**» della Axiom, si possono convertire gli oggetti in qualsiasi altro formato tridimensionale (ad esempio quelli di «Sculpt/Animate 4D», di «VideoScape 3D», di «LightWave 3D» etc.).

«PoNGo» non è un ray tracer, bensì un programma di modellazione: consente pertanto di caricare un oggetto o un group definito tramite l'editor di «Imagine», e genera a sua volta uno o più oggetti nello stesso formato.

Tutti i parametri di superficie (colore, texture, brush map, etc.) vengono mantenuti inalterati durante le trasformazioni: «PoNGo» modifica solo la posizione dei punti che definiscono il solido di partenza, cambiandone la forma. Non sono ovviamente supportate le primitive grafiche di «Imagine» quali le sfere geometriche, i path ed i «Form Object».

Il programma viene distribuito su due dischetti ed è installabile su hard disk tramite un'utility inclusa nel pacchetto. «PoNGo» utilizza, oltre che la libreria matematica standard «MathTrans.library», anche la libreria di pubblico dominio «ReqTools.library» di Nico François, autore dell'ormai diffusissimo compattatore «**PowerPacker Professional 4.0B**».

Essendo «PoNGo» fondamentalmente un programma di calcolo che lavora su matrici tridimensionali, e che quindi comporta un numero elevato di operazioni matematiche, l'uso di un Amiga dotato di scheda acceleratrice migliora notevolmente le prestazioni.

Se il programma viene lanciato da WorkBench o da CLI senza specificare parametri, verrà allocata una matrice in grado di

contenere fino a 10000 punti ed altrettanti lati. È comunque possibile modificare questo valore digitando il comando «PoNGo» da Shell seguito da un argomento numerico indicante il numero di punti da allocare.

La quantità di memoria minima richiesta da «PoNGo» è di 1Mb, possibilmente di Chip ram; per trasformare oggetti costituiti da più di 10000 punti, la quantità di memoria disponibile deve proporzionalmente essere maggiore.

«PoNGo» è stato sviluppato seguendo le direttive di programmazione Commodore ed è, di conseguenza, pienamente compatibile con tutte le versioni di KickStart e WorkBench, dalla 1.2 alla 2.0.

IL PANNELLO PRINCIPALE

Al suo avvio, «PoNGo» apre uno schermo in alta risoluzione (640 per 512 punti) contenente quattro finestre indipendenti (Fig. 1). Tre delle finestre sono usate per rappresentare l'oggetto da elaborare, mostrandolo da altrettanti punti di vista (**TriView**).

La window in alto a sinistra contiene il pannello principale di controllo, tramite il quale si effettuano tutte le operazioni sui file e la gestione degli eventi; l'interfaccia utente è molto curata ed intuitiva.

Una sessione di lavoro consiste principalmente nella realizzazione di un progetto (**Project**), ovvero nella creazione di un'animazione formata da uno o più **eventi**.

Sono possibili animazioni comprendenti fino a cinquanta eventi indipendenti, con un numero massimo di 9999 fotogrammi; se il numero di trasformazioni possibili non fosse sufficiente (un'eventualità comunque assai rara), è possibile spezzare la sequenza animata in più parti da processare separatamente in riprese successive.

Il progetto viene controllato da tre bottoni situati nella parte superiore del pannello: **New** è usato per iniziare un nuovo progetto e **Load/Save** per l'accesso ai file. Nella struttura del progetto vengono memorizzati il nome del file oggetto caricato in memoria, i parametri di visualizzazione e gli eventi (trasformazioni) programmati; nel pannello sono indicati il nome del progetto, il numero di fotogrammi della sequenza, il fotogramma corrente ed il nome del file oggetto. La parte centrale della finestra contiene i gadget per caricare o salvare un file oggetto TDDD ed un ulteriore bottone per il redraw (automatico) del solido in memoria. Segue poi una serie di comandi utili per definire i parametri di visualizzazione (come **Zoom**, **Pan**, etc.), per lanciare una sequenza animata e per aprire un requester contenente informazioni sull'oggetto o sulla memoria allocata.

GLI EVENTI

Il requester **Display** (Fig. 2) comprende la funzione **Auto Center**, molto utile per visualizzare completamente un oggetto all'interno del TriView centrandolo automaticamente nelle finestre, ed un comando (**Color**) che attiva un requester con la Palette, per modificare i colori dello schermo.

È possibile elaborare animazioni o lavorare unicamente su immagini statiche: in entrambi i casi, occorre definire gli eventi da applicare all'oggetto prescelto.

Un evento consiste in una trasformazione tridimensionale effettuata sull'oggetto correntemente in memoria: ogni evento ha una durata definita da tre variabili, che indicano il fotogramma iniziale, quello finale, e la sua velocità. Le velocità di esecuzione possibili sono: lineare (senza variazioni), accelerata, decelerata, o variabile (con accelerazione iniziale e decelerazione finale).

Sono presenti tre bottoni per il controllo di un dato evento contrassegnati con «**L**», «**R**», «**E**» ed un'icona raffigurante un occhio (Fig. 2); il primo bottone, «L», definisce se l'evento deve essere di

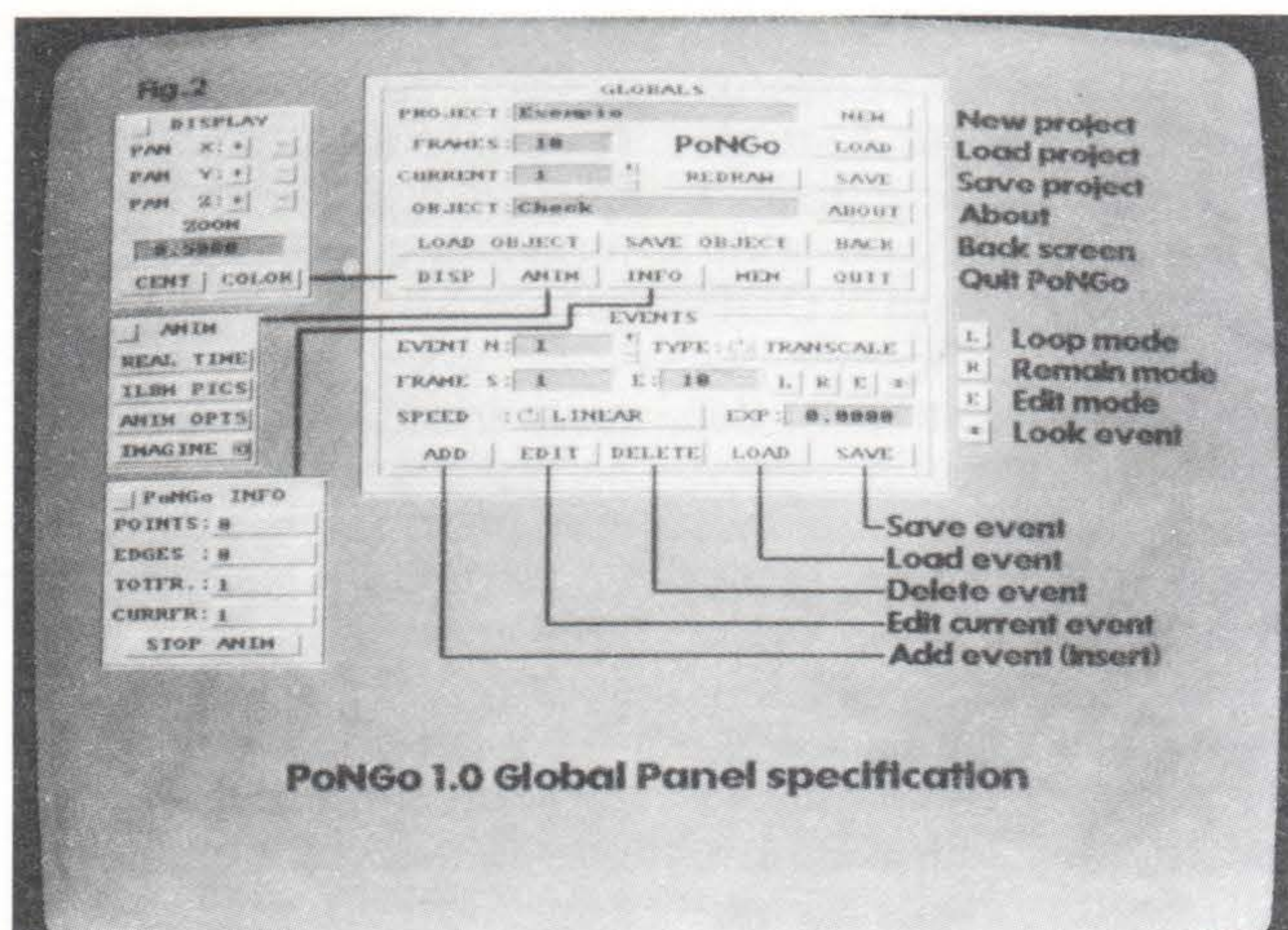



Fig. 2: il pannello di controllo principale racchiude tutte le funzioni per la gestione degli eventi e delle animazioni.

tipo **Loop** (ciclico) o meno. Se il gadget viene attivato, sarà possibile effettuare animazioni cicliche definendo particolari valori ad alcuni parametri della trasformazione: ci sono due string gadget dove vengono indicati il fotogramma iniziale e quello finale di un dato evento.

La velocità di un evento viene controllata da due gadget, che ne definiscono il tipo (**Linear**, **Speed Up**, **Speed Down**, **Oscillate**) ed il valore (**Exponent Value**).

I comandi nella parte inferiore del pannello principale vengono utilizzati per la gestione degli eventi: **Add** per inserire un nuovo evento, **Edit** per modificare la trasformazione selezionata, **Delete** per cancellare un evento, **Load** e **Save** per il caricamento ed il salvataggio di un file «.Event».

Occorre prestare attenzione al tipo di evento caricato, in quanto il numero di fotogrammi della sequenza deve essere superiore al fotogramma massimo definito dall'evento in memoria.

TRASFORMAZIONI TRIDIMENSIONALI

Con l'espressione «trasformazione tridimensionale» si intende qualsiasi operazione che modifichi la posizione dei punti che costituiscono un oggetto 3D. «PoNGo» permette di realizzare animazioni cicliche con molta facilità.

Come abbiamo già detto, durante la trasformazione tutti i parametri relativi ai solidi (lati, facce, etc.) e le loro caratteristiche di superficie (colore, texture, indice di riflessione ...) rimangono inalterati. È quindi preferibile definire tutte le proprietà relative agli oggetti che verranno trasformati da «PoNGo» prima di iniziare le trasformazioni.

Per operazioni particolari, ad esempio **Texture Animation**, si può procedere in maniera inversa attribuendo all'oggetto le caratteristiche desiderate dopo averne calcolato la trasformazione, anche se ovviamente in questo caso i tempi per la realizzazione di una sequenza completa aumentano considerevolmente.

Le trasformazioni vengono effettuate da uno o più eventi definiti nel pannello principale di «PoNGo». La versione 1.0 consente nove differenti trasformazioni, eventualmente combinabili tra loro: **Transcale** (traslazione e riduzione in scala), **Taper**, **Shear**, Bend, **Rotate**, **Twist**, **Waves** e **Metamorph** (metamorfosi).

Per attivare una trasformazione, la si seleziona con il gadget **Type** del pannello principale e si passa all'editing delle sue caratteristiche mediante il comando **Edit**.

Vediamo ora in dettaglio i singoli eventi ed i loro effetti:

TRANSCALE

L'evento **Transcale** racchiude due trasformazioni elementari: la traslazione lungo uno o più assi coordinati e la riduzione in scala sempre secondo le componenti cartesiane. Selezionato il comando Edit, «PoNGo» mostra una nuova finestra (Fig. 3) nella quale vengono impostati i parametri necessari per la realizzazione della trasformazione.

Nella parte superiore si trovano tre gadget stringa che definiscono rispettivamente i fotogrammi iniziale, quello finale e quello corrente dell'evento; questi stessi gadget sono presenti in tutti i pannelli per l'editing degli eventi.

Segue quindi un gadget per la selezione del tipo di trasformazione: si possono infatti effettuare una riduzione in scala, una traslazione, o combinare le due trasformazioni contemporaneamente. Terminato l'editing, si può effettuare un preview per poi tornare nel pannello principale.

Fig. 4: l'effetto Bend è una particolare trasformazione che «piega» un solido lungo un determinato asse.

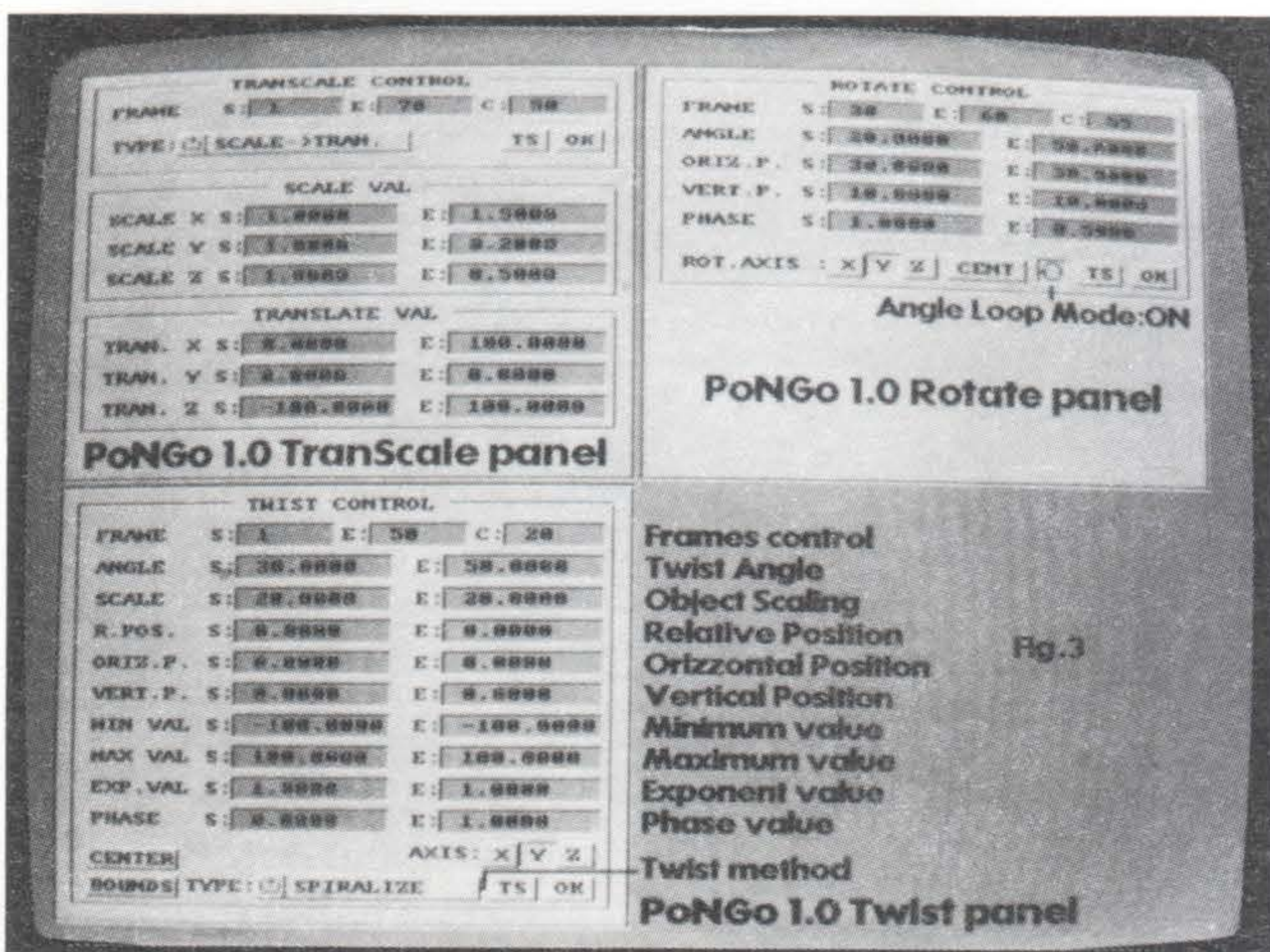



Fig. 3: i requester per le trasformazioni Transcale, Rotate e Twist (rotazione differenziata lungo un asse).

TAPER

Il **Taper** (Fig. 5) è una trasformazione tridimensionale simile allo scaling di un oggetto, con la differenza che i valori di scala variano lungo un asse prefissato.

I parametri presenti vengono utilizzati per definire la «forma» della trasformazione sull'oggetto; le differenti modalità di applicazione del Taper vengono definite da uno specifico gadget.

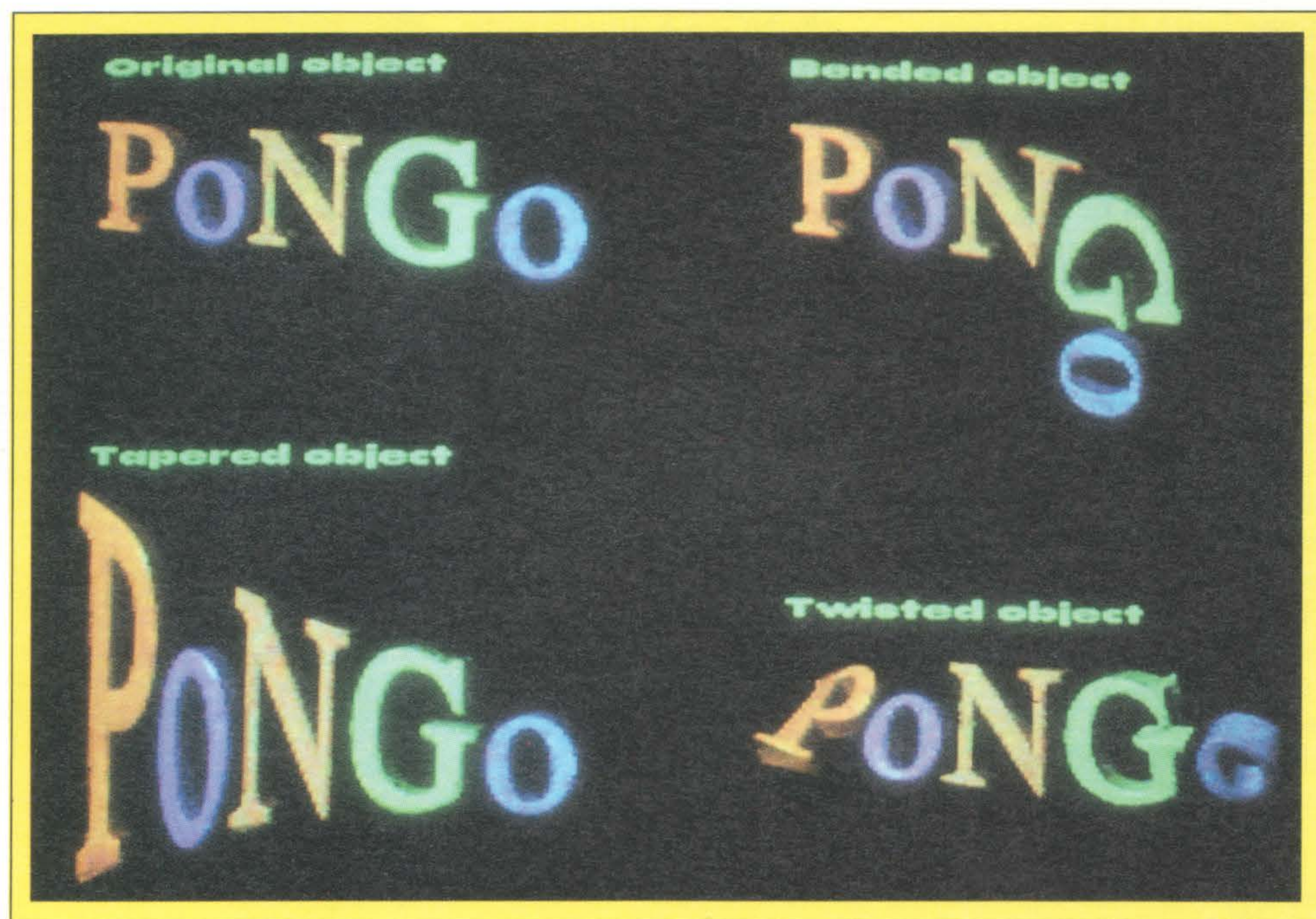
Di default, la trasformazione è di tipo **Null** (nulla); con il mouse, è possibile selezionare quattro differenti tipi di Taper: *Standard*, **Absolute** (i punti vengono considerati in valore assoluto), e **Sinusoid 1** e **2** (viene utilizzata una funzione trigonometrica). Occorre definire quindi l'asse di applicazione del Taper (con il gadget **Axis X**, **Y** e **Z**) ed eventualmente effettuare un test della trasformazione (con il gadget **TS**).

SHEAR

Lo **Shear** (Fig. 5) è, come il Taper, una trasformazione che dipende dalla posizione dei punti del solido rispetto ad un determinato asse. La differenza sostanziale sta nel fatto che in questo caso i valori restituiti dalla funzione di trasformazione vengono utilizzati per traslare i punti del solido e non per effettuarne una riduzione in scala.

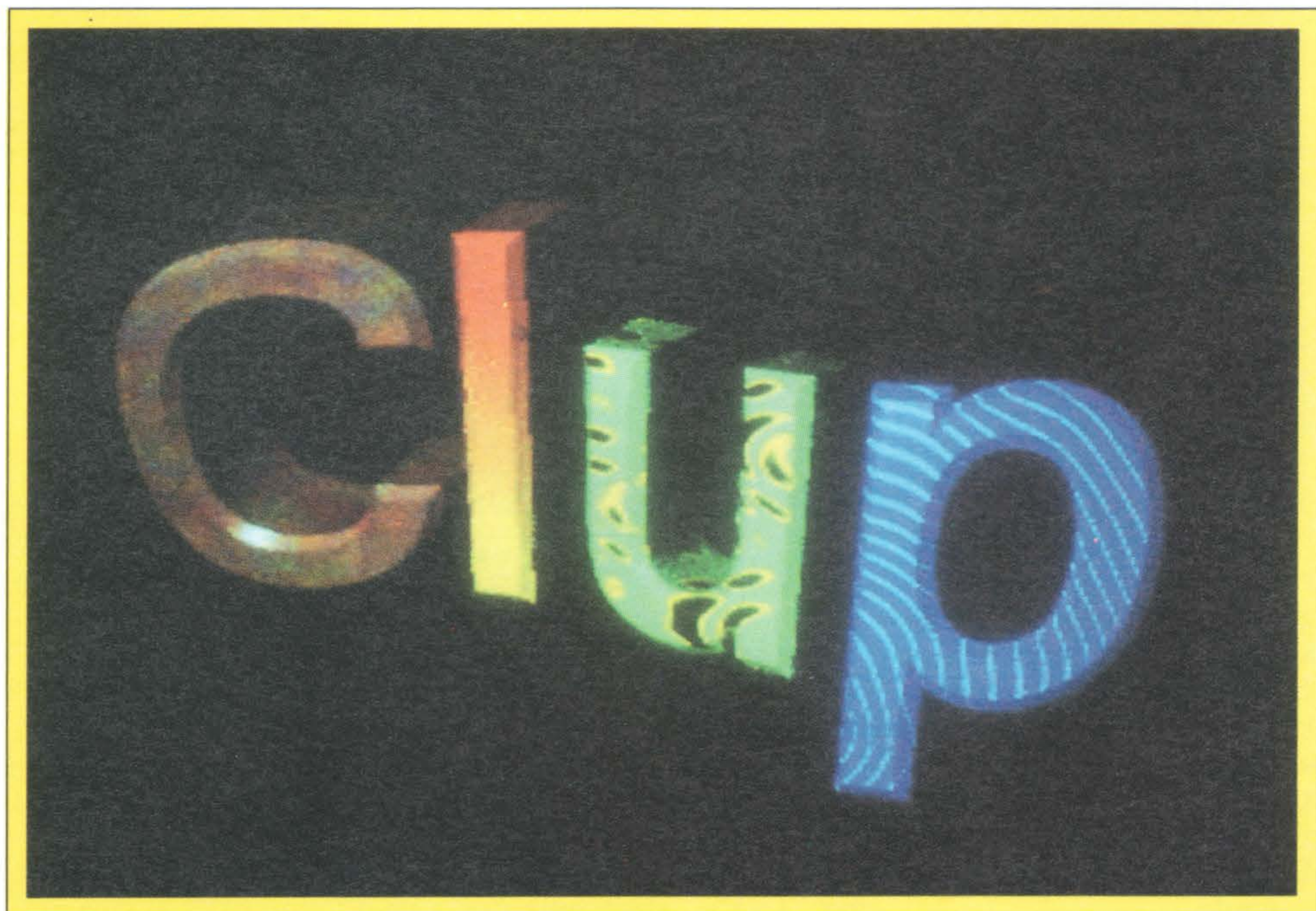
Il pannello di editing comprede gli stessi gadget del pannello Taper, per definire i parametri, la modalità di trasformazione, e l'asse di applicazione.

ROTATE

La rotazione (**Rotate**) di un oggetto può essere effettuata lungo i tre differenti assi cartesiani: basta specificare l'angolo di rotazione ed eventualmente la posizione dell'asse nei confronti dei due restanti. La posizione dell'asse di rotazione è visualizzabile attraverso il tasto **Cent** (Fig. 3).

È inoltre possibile selezionare il **Loop Mode**, relativo all'angolo di rotazione, per effettuare animazioni cicliche. Questo si rende necessario perché, durante un'animazione di questo genere, un angolo di 360 gradi produce lo stesso effetto di un angolo di zero gradi, causando la generazione di due fotogrammi coincidenti.

Attivando il gadget **Loop**, che raffigura un circolo, e specificando ad esempio un angolo finale di 360 gradi, «PoNGo» farà in modo che nell'ultimo fotogramma della sequenza l'angolo sia in realtà lievemente minore, evitando l'inconveniente.

TWIST

L'effetto **Twist** produce una «torsione» del solido, trattandosi di una particolare rotazione attorno ad un asse con un angolo di rotazione non costante, ma

dipendente dalla posizione di ogni punto appartenente all'oggetto presente in memoria. Nella finestra di editing (Fig. 3) vengono specificati l'angolo di Twist, ovvero la rotazione minima attuata nell'origine delle coordinate, ed una serie di parametri che definiscono la trasformazione. Sono disponibili cinque modalità di applicazione della trasformazione: i consueti **Standard**, **Absolute**, **Sinusoid 1** e **2**, oltre a **Spiralize**, che genera un effetto spirale molto interessante. Così come per le altre trasformazioni, occorre definire almeno uno dei tre assi cartesiani X,Y o Z.

BEND

Il **Bend** è una particolare trasformazione che «piega» un solido lungo un asse. Questo particolare effetto (Fig. 4) è molto utilizzato nei programmi di modellazione tridimensionale (con «Imagine» viene adoperata in genere la funzione **Conform to Path** a questo scopo), ma nella maggior parte dei casi la trasformazione è statica e permanente.

«PoNGo» è attualmente l'unico programma su Amiga in grado di gestire un Bend dinamico durante un'animazione.

Il primo parametro utilizzato nella trasformazione è l'angolo di Bend, espresso in gradi, che può assumere un qualsiasi valore; le successive variabili definiscono il centro e la forma della trasformazione.

L'effetto Bend può essere applicato all'oggetto in memoria secondo quattordici modalità differenti, in grado di produrre un'infinità di deformazioni. Anche in questo caso, è necessario definire l'asse di Bend prima di lanciare un preview.

WAVES

Per la simulazione di particolari effetti naturali, possono tornare utili alcune funzioni matematiche tridimensionali: ad esempio, per la simulazione di una superficie fluida (uno specchio d'acqua) o in movimento ciclico (una bandiera), vengono spesso impiegate le funzioni periodiche. Usato attentamente, l'operatore **Waves** è in grado di generare deformazioni incredibilmente fluide.

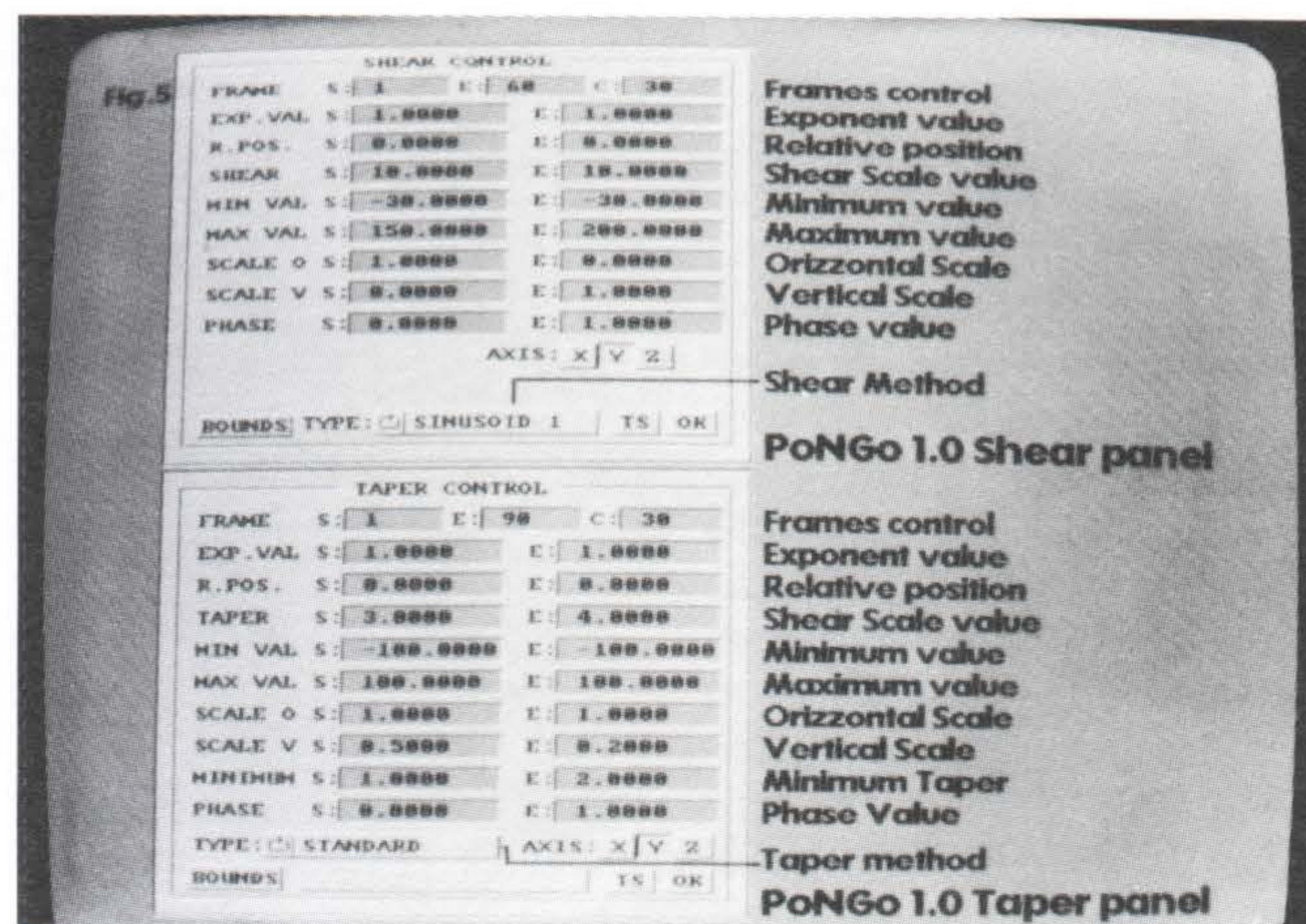



Fig. 5: i cambiamenti di posizione e di scala differenziati (Shear e Taper) sono controllati da due pannelli dedicati.

«PoNGo» comprende quaranta differenti funzioni matematiche che possono venire applicate ad un oggetto; per effetti particolarmente complessi e realistici, si possono combinare fino a cinque funzioni contemporaneamente (Fig. 6).

Le modalità di applicazione delle onde sono ben diciotto, ognuna con caratteristiche singolari: questa trasformazione molto particolare utilizza infatti una grande quantità di variabili e di funzioni matematiche.

Per facilitare la visualizzazione delle funzioni tridimensionali, sul dischetto di «PoNGo» è presente un programma specifico, «**PoNGo Waves**», che si avvale delle stesse funzioni matematiche e degli stessi parametri, con l'unica limitazione che si possono combinare fino ad un massimo di tre funzioni contemporaneamente.

In «PoNGo Waves» le onde vengono applicate linearmente ad un piano di dimensioni definite dall'utente e sono possibili animazioni in tempo reale; viene anche fornita una semplice utility di sola visualizzazione delle funzioni d'onda.

METAMORPH

La metamorfosi di un solido è probabilmente una delle trasformazioni più affascinanti, anche se difficile da realizzare. «PoNGo» consente addirittura la metamorfosi multipla di più solidi nel corso di una sequenza animata.

Nel pannello di controllo (Fig. 6) della metamorfosi, l'utente definisce l'oggetto finale attraverso il file requester ed il programma calcolerà la trasformazione tra l'oggetto in memoria e quello definito come desti-

Fig. 6: «PoNGo» è in grado di applicare una o più funzioni d'onda ad un solido e di effettuare metamorfosi multiple.

nazione. «PoNGo» permette la metamorfosi tra oggetti aventi diversa struttura, anche se in questo caso i risultati possono essere imprevedibili.

La possibilità di effettuare metamorfosi multiple anche di oggetti con struttura differente è una delle caratteristiche uniche di questo programma.

GESTIONE DELLE ANIMAZIONI

«PoNGo» è in grado di effettuare trasformazioni su di un singolo oggetto o in animazione su più solidi: osservando infatti tutti i pannelli di controllo delle trasformazioni, la maggior parte dei parametri ha un valore iniziale ed uno finale.

Prima di iniziare la sequenza, «PoNGo» apre un pannello in cui l'utente specifica il tipo di animazione, decidendo se mostrarla in tempo reale (**realtime**) o se generare singole **immagini IFF**, animazioni **compresse** (nel formato IFF ANIM 5, supportato da «Deluxe Paint» etc.) oppure **oggetti** nel formato **TDDD** (Three Dimensional Data Description) di «Imagine» e «TurboSilver».

Nel pannello principale, per ogni evento, si possono definire gli incrementi delle variabili dinamiche variando quindi la velocità dell'evento. Sono possibili variazioni lineari (**Linear**), accelerate (**Speed Up**), decelerate (**Speed Down**), o una combinazione di queste ultime due (**Oscillate**).

Un parametro esponenziale consente il controllo di queste ultime variazioni.

Abbiamo già detto come «PoNGo» sia attualmente l'unico programma in grado di effettuare le trasformazioni sopra elencate all'interno di una sequenza animata. Alcuni modellatori come «Imagine 2.0», «Real3D 1.4», «3D Pro 2.0» ed il Modeler di «LightWave 2.0» dispongono di alcune delle trasformazioni (Bend, Taper e Twist) presenti in «PoNGo», ma nessuno di questi ottimi programmi è in grado di gestirli dinamicamente in animazioni.

COMBINAZIONE DI EVENTI

La grande potenza di «Pongo» sta nella possibilità di combinare fino a cinquanta eventi, mettendo a disposizione una grande varietà di trasformazioni multiple. Il numero di parametri presenti per ogni trasformazione è stato definito in modo tale da rendere gli eventi completamente gestibili dall'utente.

INTERFACCIAMENTO CON IMAGINE

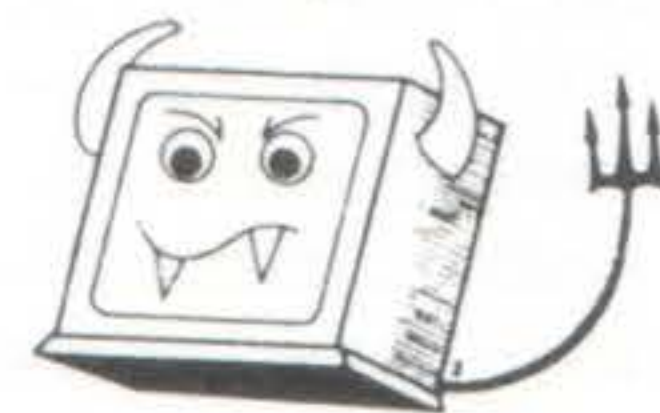
La simulazione di oggetti animati e di alcuni fenomeni naturali è possibile grazie a queste trasformazioni; utilizzando eventi con differenti velocità, si possono ottenere effetti di grande realismo. Le vere potenzialità del programma possono essere messe in luce con un uso attento e mirato al particolare effetto. Come ultima precisazione, è importante tener presente che, una volta generati gli oggetti, entrati nell'Action Editor di «Imagine» e definiti i file di uno o più «Actors», sono disponibili gli effetti (F/X) del programma e tutte le altre trasformazioni (elementari) presenti all'interno dell'editor stesso come lo Scaling, lo Zoom, e così via.

SUL DISCHETTO

L'interfacciamento con «Imagine» risulta quindi semplice ed efficace in quanto tutte le potenzialità dell'Action Editor del programma rimangono inalterate.

Sul dischetto di «PoNGo» è presente inoltre un'utility in grado di modificare automaticamente il file *Staging* di «Imagine»: in questo modo è semplice inserire automaticamente la sequenza generata da «PoNGo» in un progetto di «Imagine» già esistente.

Gli oggetti creati da «PoNGo» potranno essere agganciati ad un path, esplodere o trasformarsi in base ad un qualsiasi F/X esattamente come qualsiasi oggetto generato direttamente dall'editor interno di «Imagine».

□

The best of the PD

Un servizio davvero unico per i lettori di AmigaByte, che sceglie per voi i migliori programmi fra quelli di Pubblico Dominio esistenti che, per ragioni di spazio, non possono essere inseriti nel dischetto allegato alla rivista, e li riunisce in dischetti esclusivi, uno più interessante dell'altro.

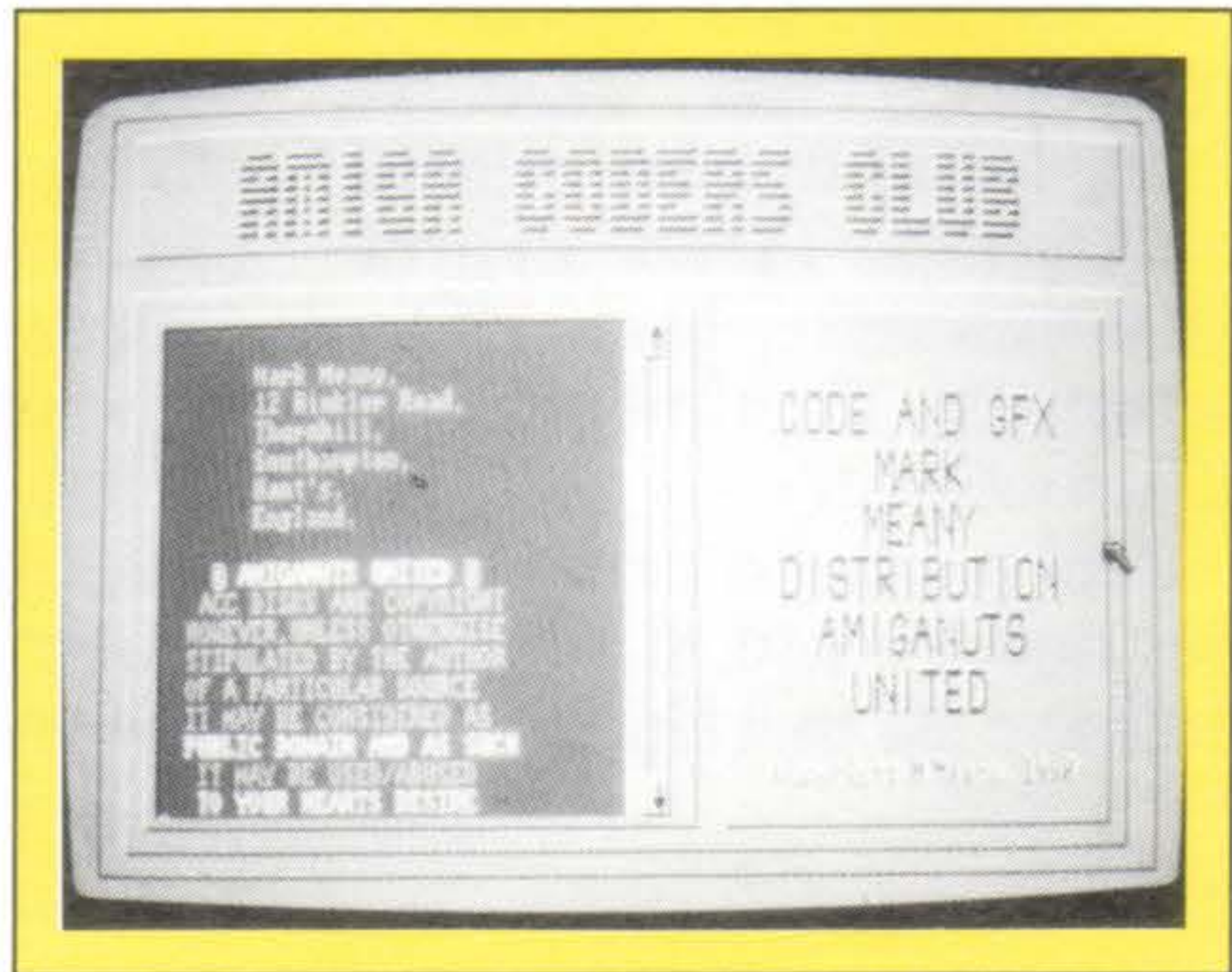
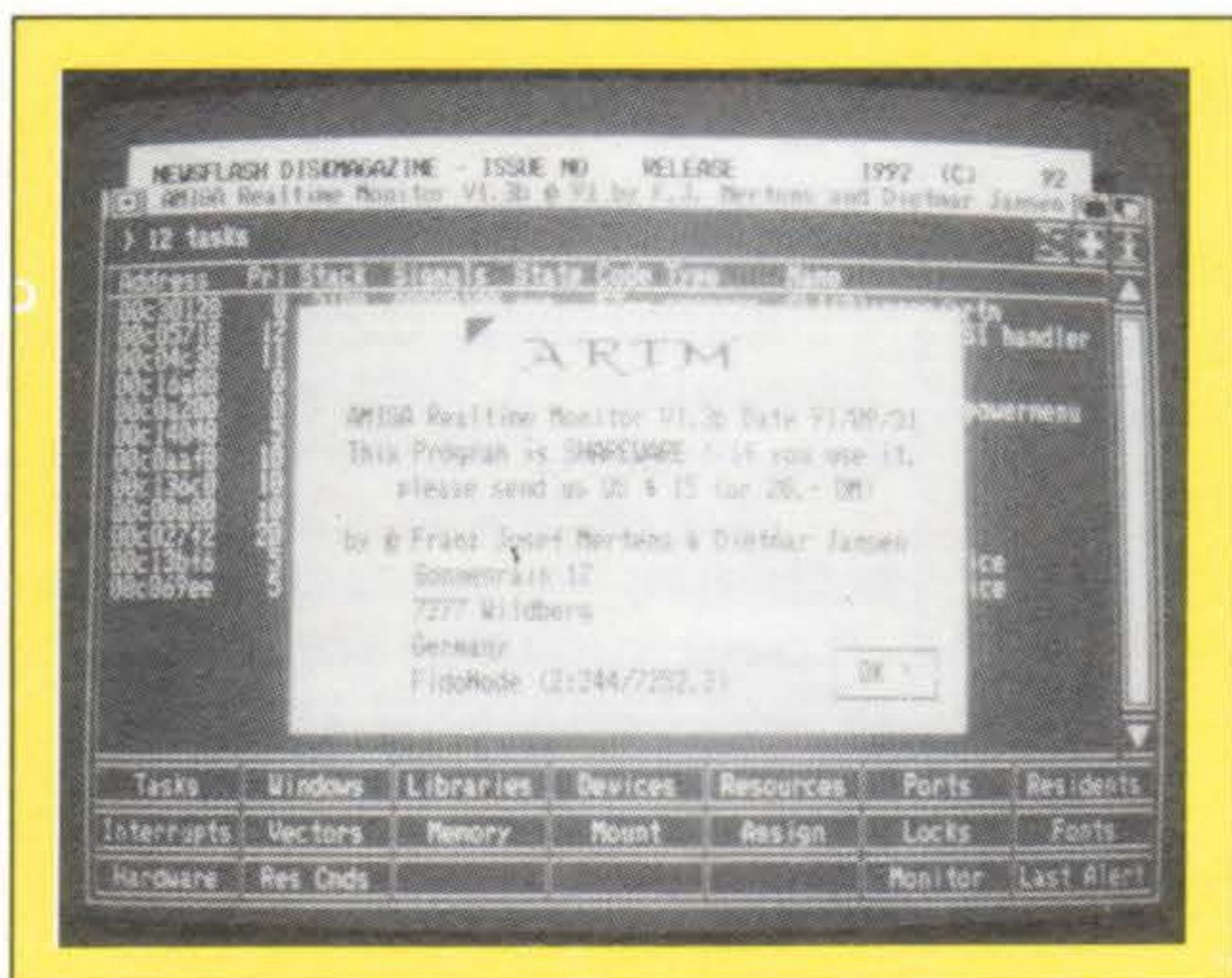
di EMANUELE SCRIBANTI

NEWSFLASH 23

Questo attuale numero della rivista su disco più diffusa in Europa arriva, come il precedente, su tre dischi; fra i testi segnaliamo uno speciale sulle ultime novità software dedicate alla produzione video, e ben due articoli sul musicista francese Jean Michel Jarre, indubbiamente uno dei protagonisti di questo numero; non mancano, naturalmente, le consuete recensioni hardware/software.
La directory dei programmi non è meno interessante: oltre che «KeymapEd» e «Degrader» (apparsi anche su AmigaByte 38), vi si può trovare una nuova versione (la 1.3b) di «ARTM», una delle utility più complete in circolazione, imperdibile per chiunque voglia fare conoscenza con il proprio computer ad un livello più approfondito di quanto permetta l'interfaccia utente.
Con «Amiga Real Time Monitor» (è questo il significato della sigla) potete sbirciare l'elenco dei task in esecuzione, congelandone o eliminandone alcuni, o modificando le priorità a vostro piacimento; sapere quale programma tiene aperto un determinato file; visualizzare l'elenco delle finestre e degli schermi attivi, ed eventualmente chiuderne qualcuno; verificare quali e quante siano le librerie di funzioni in uso, ciascuna con il numero di task che vi sta accedendo; visualizzare i comandi residenti, e molto altro ancora.
Avrete capito che «ARTM» si rivela insostituibile nello sviluppo e nel debugging dei vostri programmi, e per ricercare le cause di malfunzionamento del sistema.
La collezione di programmi di questo mese è completata da «Decode» (uno strumento che aiuta la codifica e la decodifica di file contenenti testo, come i dati di quell'avventura che proprio non riuscite a risolvere...); Zoom 5.3 (uno dei più usati compattatori per dischi, ora con interfaccia grafica ed un mucchio di nuove opzioni); «DirWork» (una directory utility dello stesso genere di «DirOpus» e «SID», configurabile e piuttosto potente); «DiskRepair», l'ultimo grido in fatto di restauratori di dischi danneggiati, caratterizzato sfortunatamente da istruzioni per l'uso soltanto in tedesco.
Come d'abitudine, questo ultimo numero di NewsFlash comprende anche un paio di font inediti in formato IFF, uno dei quali davvero molto bello, e delle demo in linguaggio macchina (questo mese ben tre: segnaliamo «Arcade Gaemo», che include un mini-shoot'em up a scorrimento orizzontale). Sono inoltre presenti una raccolta di clip art, cioè di immagini in bianco e nero in alta risoluzione, che raffigurano in prevalenza animali, da usare con programmi di DeskTop Publishing, e tanta, tanta musica: cinque moduli sulla rivista, più due sul dischetto supplementare. Questi ultimi, in particolare, sono realizzati con tanta cura nei campionamenti da risultare lunghi la bellezza di 400 Kb ciascuno.
Con l'eccezione del terzo disco, la rivista gira anche su Amiga senza espansione di memoria; segnaliamo però che questo numero, purtroppo, a differenza dei precedenti, non è totalmente compatibile con Amiga 500 Plus.

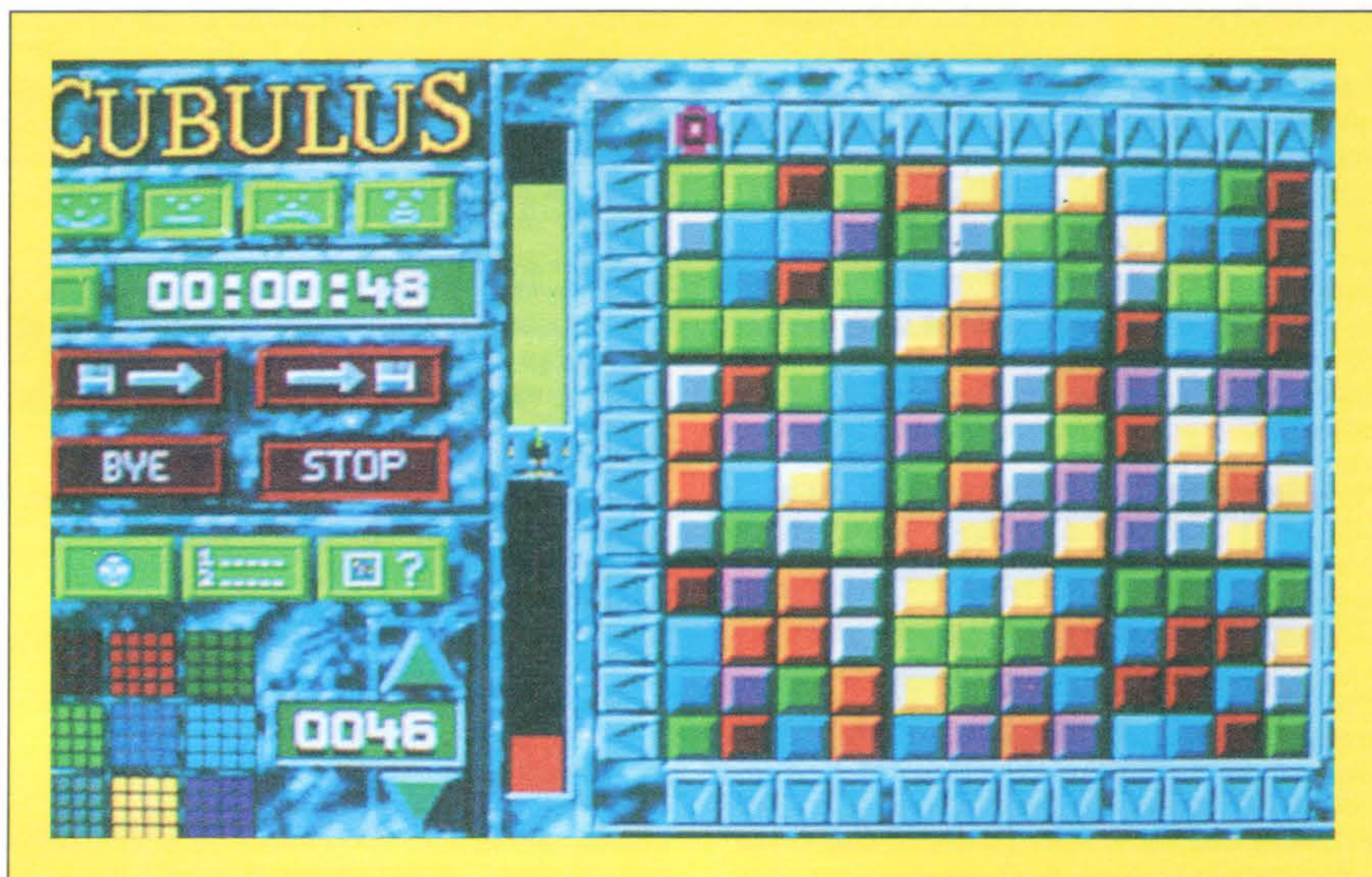
Disponibile sui dischetti: NewFlash 23 (tre dischi, 22.000 lire)

AMIGA CODERS CLUB 19

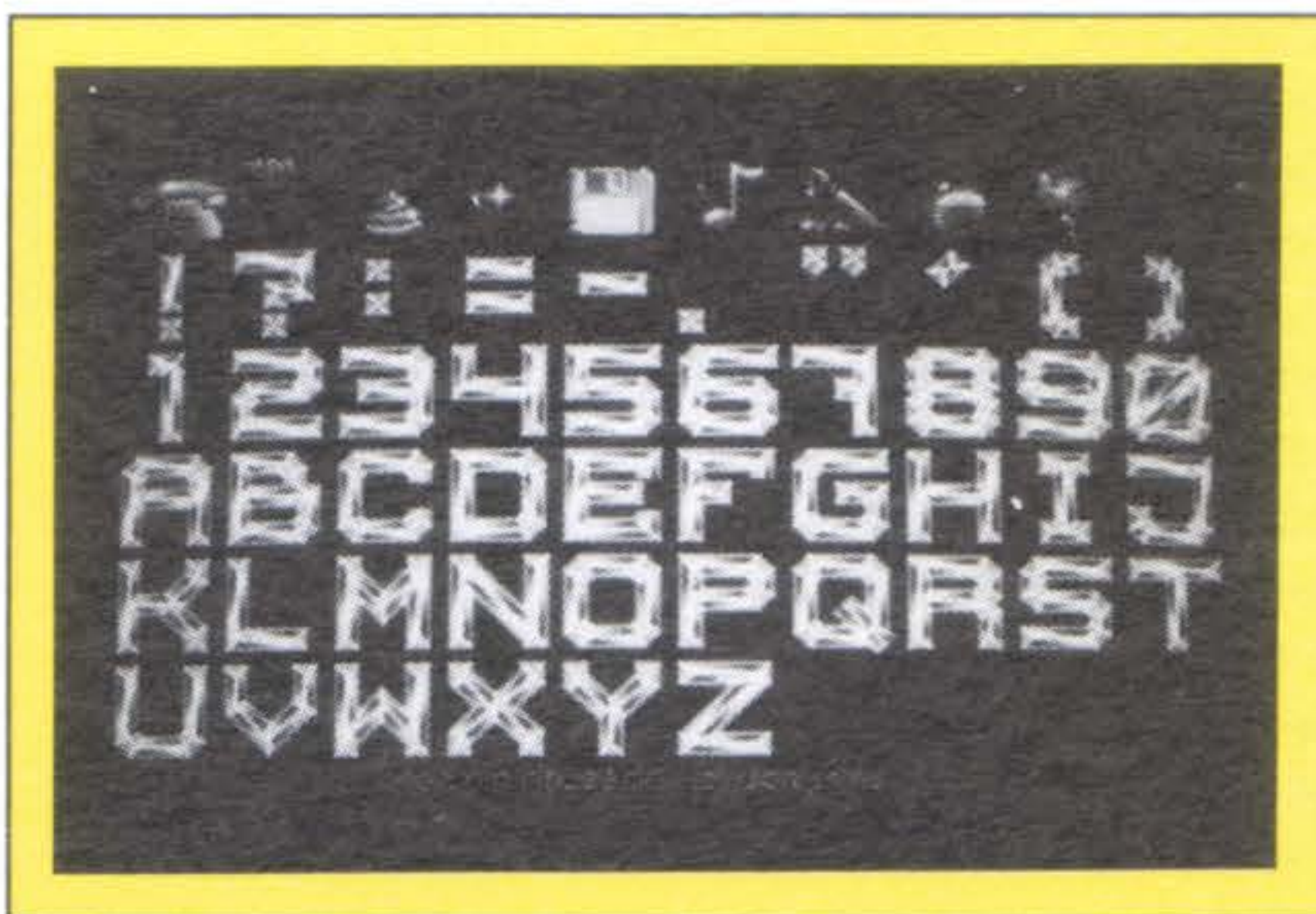
Prosegue a gonfie vele la navigazione di questa rivista inglese su dischetto, un supporto davvero insostituibile per chiunque programmi in Assembly, a qualsiasi livello. Lo spazio riservato agli eseguibili è molto ridotto, in confronto alla gigantesca mole di sorgenti offerti in pasto ai voraci lettori.
Gli argomenti trattati sono, come è ovvio, molteplici: «ReadKey» dimostra un sistema rapidissimo, anche se non troppo ortodosso, per la lettura della tastiera: «Program Selector» mostra come sia possibile programmare un menu che si occupi della selezione dell'avvio di programmi.
Sul versante della grafica, «Lines» e «FillLines» sono esempi di come generare grafica vettoriale con la velocità tipica del linguaggio Assembly, mentre «Bobs» sfoggia come animare, con il valido aiuto del Blitter, veri e propri oggetti multicolori disegnati da voi.
«Module Loader» è un esempio di lettura e di esecuzione di moduli SoundTracker; «Intro Source» è il sorgente di una spettacolare dimostrazione delle capacità di Amiga.
Per quanti non si limitano a programmare intro e demo, ma ambiscono a qualcosa di più evoluto, ecco un sorgente completo per la valutazione di espressioni algebriche, accompagnato da una puntuale descrizione dell'algoritmo utilizzato: è sicuramente meno potente della vostra calcolatrice scientifica, ma costituisce comunque un'ottima base per realizzazioni di questo tipo.
Il dischetto contiene infine «PowerSource», una recente e praticissima utility liberamente distribuibile, con la quale programmare finestre e gadget diventa davvero un gioco da ragazzi: dopo averli disegnati sullo schermo, penserà «PowerSource» a generare il codice corrispondente, in Assembly oppure in C.
Tutti i sorgenti sono compressi con «PowerPacker»: un'apposita opzione del menu si occupa della scompattazione, necessaria per chi desiderasse compilare il materiale pubblicato; i sorgenti in forma non compressa occupano la bellezza di due dischi. In alternativa potete utilizzare il programma «PP» (PowerPacker Patch) per

caricarli direttamente nel vostro assembler preferito.
La rivista può essere consultata senza problemi anche dagli utenti di A500+ con Kick 2.0; alcuni dei programmi pubblicati, però, potrebbero rifiutarsi di funzionare. Mark Meany, direttore della rivista, assicura comunque che l'inconveniente è stato rimosso a partire dal numero 20, selezionando i sorgenti in modo che girino perfettamente anche sul nuovo modello.

Disponibile sul dischetto: ACC 19 (lire 12.000)

CUBULUS

Eccezionale: per la prima volta è disponibile, con la formula del pubblico dominio, un metodo sicuro per impazzire. Più raffinato del suicidio, più rapido di tutti i sistemi finora conosciuti, si chiama Cubulus ed ha le innocue sembianze di un passatempo: in realtà, i quadratini colorati che riempiono lo schermo hanno l'unico scopo di dissimulare la pericolosità di questo prodotto.
Il campo di gioco è un quadrato, suddiviso in «piastrelle» (in numero variabile a seconda del livello di difficoltà), di colori diversi. Ciascuna di queste è a sua volta divisa in sedici quadratini, dando origine ad una configurazione come quella mostrata in figura.
Attivando con il mouse le freccine ai bordi dell'area, possiamo far scorrere una riga o una colonna di quadratini: quelli che escono rientreranno dal lato opposto. Con il pulsante sinistro muoveremo una sola riga, mentre con il destro agiremo su tutte le quattro che compongono la fila di piastrelle selezionata.
Avrete notato che il gioco presenta numerose analogie con un cubo di Rubik: in entrambi, infatti, abbiamo dei colori da riordinare. Ma mentre nel malefico solido ungherese la difficoltà derivava dalle tre dimensioni, in questo caso la fonte dei vostri problemi è la necessità di dover riordinare i quadratini all'interno di ogni piastrella, ed in seguito le piastrelle stesse: se all'inizio si può procedere «a vista» decidendo volta per volta le mosse da compiere, avanzando nel gioco si rende necessario sviluppare una strategia ben definita, pianificando in anticipo una sequenza di operazioni che consenta di ottenere il risultato desiderato.
Cercate di rimanere calmi, dunque, quando il puzzle sarà praticamente risolto «salvo per quei due quadratini fuori posto»: può darsi che la parte difficile debba ancora iniziare!
I quattro livelli di difficoltà, selezionabili per mezzo delle faccine sotto il titolo della schermata, corrispondono rispettivamente a puzzle con 4, 9, 16 o 25 piastrelle: facendo due conti, si tratta di 64, 144, 256 o 480 quadratini colorati che popoleranno i vostri peggiori incubi rifiutando testardamente di posizionarsi là dove li avete destinati.
La difficoltà è influenzata anche dal grado di «rimescolamento» del rompicapo: da 1 (ma consigliamo di partire almeno da 6/7) a 1000, per i più masochisti.
Dato che mettere un po' d'ordine in tutto questo caos può portarvi via molto tempo, gli autori hanno gentilmente inserito un'opzione per il salvataggio su disco (e relativo caricamento) della partita in corso, togliendo così l'ultima scusa rimasta per abbandonare l'impresa qualora si rivelasse eccessivamente ardua; clickando sul quadratino a sinistra del timer potrete perfino concedervi una pausa.
No, non vi illudete di poter fare i furbi approfittandone per studiarvi con calma la situazione: il piano di gioco scomparirà dalla vostra vista finché non deciderete di continuare.
La grafica ed il sonoro di «Cubulus» sono di altissima qualità, un fenomeno difficilmente riscontrabile nei giochi di pubblico dominio: gli autori, che hanno incautamente inserito i loro nomi nel programma, esponendosi così alle vendette ed alle minacce di migliaia di utenti, sono Tobias Richter e Thomas Kruza.

Disponibile sul dischetto: AmigaByte PD 108

UGA POWERFONTS 1

Dalla UGA software arriva il primo titolo di quella che ha tutte le premesse per diventare una nutrita collana: ci riferiamo alla serie «PowerFonts», una raccolta di set di caratteri memorizzati sotto forma di immagini IFF. Questo standard è largamente utilizzato da tutti quei programmi che hanno bisogno di adoperare i font a grande velocità, come i vari generatori di intro: la relativa complessità del formato standard dei font bitmap Amiga, in cui ogni carattere deve essere separato dagli altri prima dell'uso, rende impossibile il loro impiego in applicazioni di questo genere.

I font IFF sono, al contrario, direttamente accessibili tramite qualsiasi pacchetto grafico ed adattabili a qualsiasi esigenza. Per semplificare ulteriormente la vita ai possessori del Red Sector Demo Maker ed evitare conversioni, i font sono memorizzati anche nel formato particolare richiesto da quel programma.
La sigla UGA significa «United Graphic Artists»: non stupisce, pertanto, che i caratteri contenuti nel pacchetto siano realmente di ottima qualità. Ce ne sono ben dieci, ed i loro nomi costituiscono già dei validi indizi per avere un'idea della forma: Alien, Fire, Shine, Stone, Stone2, College, Crystal, Future, Square e TwinFont. Sono tutti realizzati in bassa risoluzione, a 16 o 32 colori: ed a proposito di colori, bisogna riconoscere che sono stati utilizzati davvero sapientemente per conferire ai caratteri effetti di tridimensionalità, trasparenza, lucidità, ed altro ancora.
I «PowerFonts» si possono sfruttare anche senza un programma che li supporti specificamente: caricandoli con «DeluxePaint», ad esempio, si potranno comporre a mano le scritte desiderate, prendendo ogni singolo carattere come brush e «stampandolo» là dove serve. La creazione di schermate introduttive e di suggestivi titoli per i propri programmi diventa così un gioco da ragazzi.
Il vantaggio del formato IFF consiste poi nella possibilità di modificare la palette utilizzata in modo molto semplice (è sufficiente qualsiasi programma di disegno o di elaborazione di immagini). Sempre con «DeluxePaint», l'opzione Remap è sufficiente nella maggior parte dei casi ad adattare la palette dei font a quella dei disegni sui quali si intende sovrapporli.

Disponibile sul dischetto: UGA PowerFonts I (L. 15.000)

LANDSCAPE

Il bello del software PD è che, di tanto in tanto, capita di imbattersi nei progetti più strampalati, programmi dedicati ad applicazioni per le quali, probabilmente, nessuna software house si sentirebbe di investire mezzo dollaro.
Eppure, ne siamo certi, molti di voi hanno sempre prepotentemente avvertito il bisogno

di un CAD per il disegno di giardini! Potete smettere di macerarvi nell'attesa: oggi c'è «Landscape».
I possibili utilizzi sono due: se avete già un giardino, potete provare a ridisegnarlo quanto più fedelmente possibile, e divertirvi ad apportare modifiche di qualsiasi genere senza ricorrere all'aiuto di un esperto giardiniere.
Se vivete in un condominio in centro, invece, il compito è ancora più facile, dato che la vostra fantasia non avrà alcun vincolo cui sottostare, e potrà sbizzarrirsi nella creazione del giardino dei vostri sogni.
Per prima cosa si procederà alla definizione dei contorni del giardino (Draw Boundaries); in seguito si aggiungeranno il prato (Lawn), i vialetti pedonali (Footpath), le aiuole di terra (Ground) e, naturalmente, le piante.
È qui che «Landscape» offre il meglio: alberi sempreverdi e non, in tre taglie differenti (occhio, però: quella denominata «Medium» presenta forti affinità con un baobab millenario); rosai, generiche piante con fiori (di colore definito o vario) o senza, anch'esse in diverse dimensioni. Si tratta di un aspetto piuttosto importante, in quanto solitamente i programmi di grafica vettoriale hanno difficoltà nel disegno di forme non geometriche: occorre riconoscere che «Landscape» ci riesce in modo egregio.
Quando avrete posizionato bene il tutto, salvate il progetto (non sarà possibile farlo più avanti) e selezionate «Draw Garden»; sarete invitati ad indicare la direzione di vista dell'osservatore, dopodiché il disegno avrà inizio, e con esso la crescita di una lussureggiante vegetazione.
Al termine di questa fase, che può portare via qualche minuto, avrete la possibilità di memorizzare il capolavoro sotto forma di immagine IFF, per eventuali ritocchi con «DeluxePaint» o similari (aggiunta di steccati, edifici, corsi d'acqua, talpe ed insetti nocivi e/o ripugnanti).
Se invece desiderate cambiare il punto di vista, occorrerà uscire dal programma e ricaricare il progetto (è questo, forse, l'aspetto che ci lascia più perplessi).
«Landscape» funziona soltanto con il Kickstart 1.3: i possessori di Amiga 500 Plus quindi dovranno limitarsi a progettare il giardino dei loro sogni con carta e penna.

Disponibile sul dischetto: AmigByte PD 152

BUG BASH

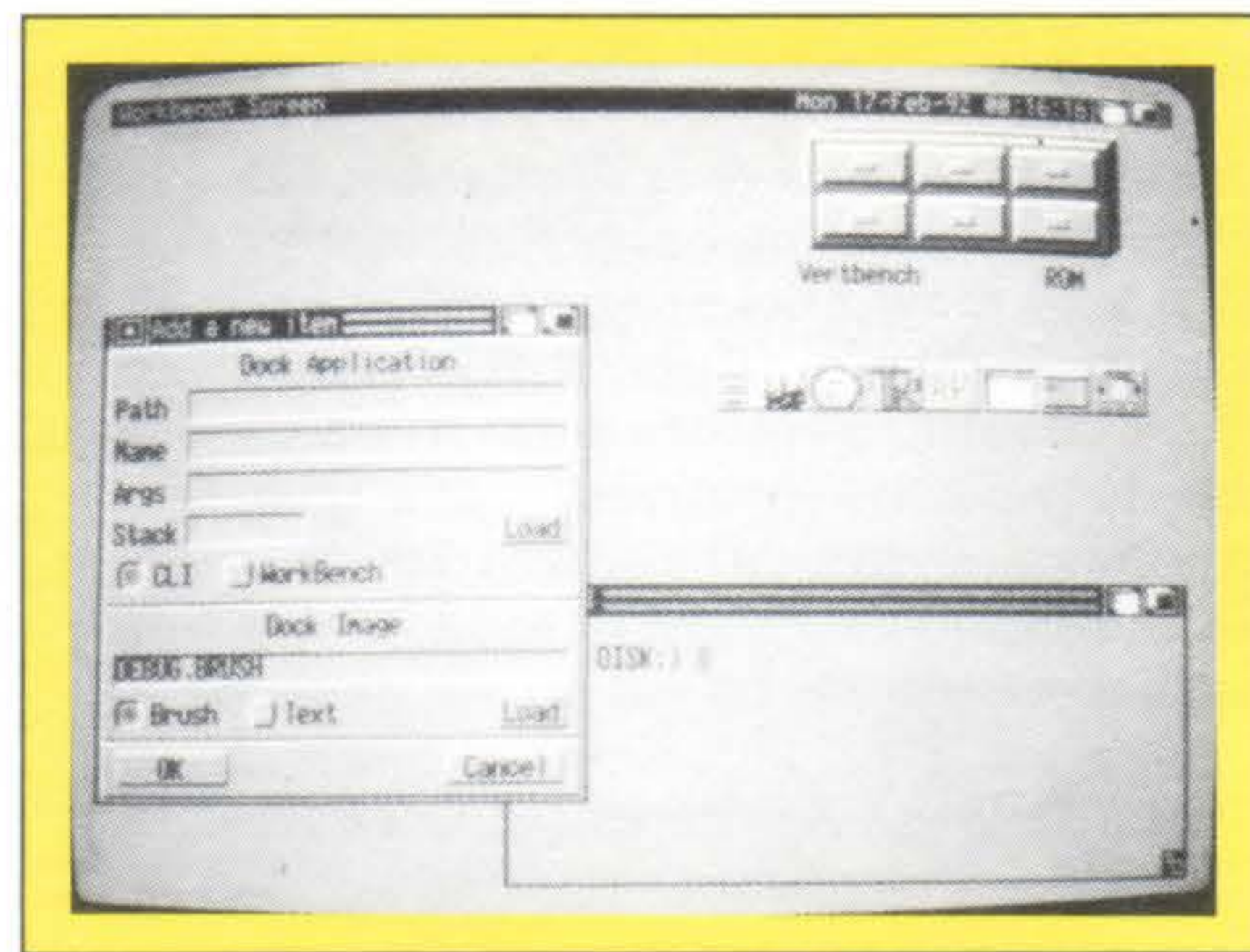
Ma chi l'ha detto che il software di cui si parla in queste pagine debba essere per forza serio e un po' noioso?
«Bug Bash», alla sua nascita, era un gioco commerciale; come spessissimo accade, cessata la distribuzione nei normali circuiti, ha avuto una seconda giovinezza grazie alla sua inclusione nel catalogo della Amiganuts United, che comprende software PD o

commerciale a basso prezzo (e di cui AmigaByte, lo ricordiamo, ha l'esclusiva per l'Italia).
Attenzione, dunque: «Bug Bash» non è liberamente copiabile e distribuibile; se ne parliamo in queste pagine è unicamente per via del suo costo.
Il nostro giardino è stato invaso nottetempo da una smisurata quantità di disgustosi insetti: armati unicamente di una bombola di insetticida, dobbiamo naturalmente distruggerli prima che la nostra energia vada a zero.
Ah, dimenticavamo un piccolo particolare: non ne conosciamo le cause, ma gli insetti in questione sono grandi quanto noi!
Si tratta di ragni, zanzare, cimici, pulci, api e falene, coloratissimi, ben disegnati e soprattutto con delle facce davvero ebeti, per togliere anche al giocatore più ecologista ogni scrupolo derivante dal pesante utilizzo di insetticida.
Poiché i malnati possono naturalmente volare, anche a noi è concessa questa facoltà, per mezzo di un buffo zaino ad elica che presenta l'unico rilevante difetto di impiegare istanti preziosi prima di avviarsi.
Già che ci siete, cercate di approfittare dell'occasione per dare una pulita al prato, ricoperto da lattine usate e da altri rottami:

la cosa non potrà che giovare al vostro punteggio.
«Bug Bash» appartiene dunque di diritto alla categoria degli «shoot-em-up» (termine con il quale si è soliti indicare quei giochi il cui unico scopo è sparare ai nemici); la sua peculiarità consiste nell'ambientazione, decisamente insolita. Il funzionamento è garantito anche con il nuovo Amiga 500 Plus.

Disponibile sul disco: Bug Bash (Lire 10.000).

AMIDOCK

Se tra i nostri lettori c'è qualche utente di computer NEXT, ciò di cui stiamo per parlare gli risulterà familiare; per tutti gli altri, è obbligatoria una breve introduzione.
Con Amiga, subito dopo il boot, solitamente ci si trova in ambiente WorkBench; se si dispone di un hard disk con molte partizioni e parecchi livelli di sotto-directory annidate l'una dentro l'altra, andare a richiamare il programma desiderato può essere una procedura piuttosto laboriosa.
Anche il caso contrario, quello in cui si lavori unicamente su floppy, non è privo di difficoltà: occorre inserire il disco corrispondente al programma da caricare, aprirlo con un doppio click, ed attivare l'icona desiderata; data la non proverbiale velocità dei drive di Amiga, anche questa

operazione può portare via più tempo del dovuto.
Se poi dovete richiamare un'applicazione priva di icona, vi tocca aprire una shell e digitare il nome del comando.
Il WorkBench 2.0 pone rimedio a questi inconvenienti, permettendo di far apparire alcune icone sul piano del WorkBench all'avvio del sistema, e di vedere i programmi privi di icona; ma sicuramente si può avere di più.
«AmiDock», derivato proprio da un'utility per il computer NEXT, apre una finestra con una serie di mini-icone, clickando le quali si può avviare qualsiasi programma.
I disegni sono memorizzati sotto forma di brush IFF (e possono dunque essere realizzati con qualsiasi programma di disegno); in più, il file Dock.config contiene il path completo dei programmi loro associati.
In questo modo, l'avvio del vostro software più usato avviene letteralmente in una frazione di secondo: basta un click con il pulsante sinistro, seguito eventualmente dall'inserimento del disco.
I programmi possono essere lanciati da WorkBench o da CLI, ed in questo secondo caso si possono specificare i parametri desiderati. I simboli sono disposti in una fila (a scelta orizzontale o verticale); se sono troppi per stare nello schermo, due freccine vi permettono di spostarvi rapidamente nell'elenco. Il pacchetto comprende una nutrita schiera di simboli preconfezionati per i programmi più diffusi, realizzati nella doppia versione per schermi interlacciati o meno; «AmiDock» richiede l'arp.library, ed è compatibile con tutti gli Amiga in circolazione, indipendentemente dalla versione di WorkBench utilizzata: una volta installato, compie il suo dovere in modo efficiente e discreto, e ben difficilmente deciderete di rimuoverlo dal vostro disco di sistema.

Disponibile sul dischetto: AmigaByte PD 152

Ogni dischetto contenente i programmi PD recensiti su queste pagine costa (salvo indicazione diversa) lire 10 mila (13 mila per riceverlo espresso). Per ricevere i dischetti basta inviare vaglia postale ordinario intestato ad AmigaByte, c.so Vitt. Emanuele 15, 20122 Milano, specificando sul vaglia il codice dei dischi (esempio: AmigaByte PD 2) indicato sulla rivista ed i propri dati completi in stampatello.

Tutto quello che c'è da sapere sugli hard disk

Acquistare un hard disk è ormai prioritario perché, tra le memorie di massa esistenti, il disco rigido rappresenta oggi l'optimum per il miglior rapporto tra costo, velocità d'accesso alle informazioni e capacità di memorizzazione.

di ANTONIO DE LORENZO

Parallelamente all'evoluzione dei personal computer e all'avvicendarsi di macchine sempre più potenti e sofisticate, capaci di trattare grandi quantitativi di dati a velocità sempre maggiori, anche i metodi di memorizzazione dei dati si sono adeguati di pari passo (dai registratori a nastri ai floppy disk e, più recentemente, ai CD ROM).

Tra le memorie di massa, gli hard disk rappresentano attualmente quelle con il miglior rapporto tra costo, velocità d'accesso alle informazioni, e capacità di memorizzazione.

Ora che l'acquisto di un **hard disk** è diventato una delle voci prioritarie (seconda forse solo a quella di un'espansione di memoria) anche per gli utenti Amiga, è arrivato il momento di saperne di più.

COM'È FATTO UN HARD DISK

L'hard disk (o disco rigido, così detto perché realizzato con supporti magnetici rigidi al contrario dei flessibili floppy) nasce intorno agli anni '60 al termine di un lungo lavoro di ricerca svolto nei laboratori della IBM.

Il disco rigido può esser considerato a tutti gli effetti un miglioramento del floppy disk in quanto si basa sui medesimi principi costruttivi. Così come per i floppy, gli hard disk presenti sul mercato sono disponibili in due formati: **5¼** e **3½** pollici; i primi però stanno progressivamente scomparendo a netto favore dei secondi, più piccoli e veloci.

Un hard disk è composto da una serie di dischi metallici impilati, ricoperti da un sottile strato di materiale

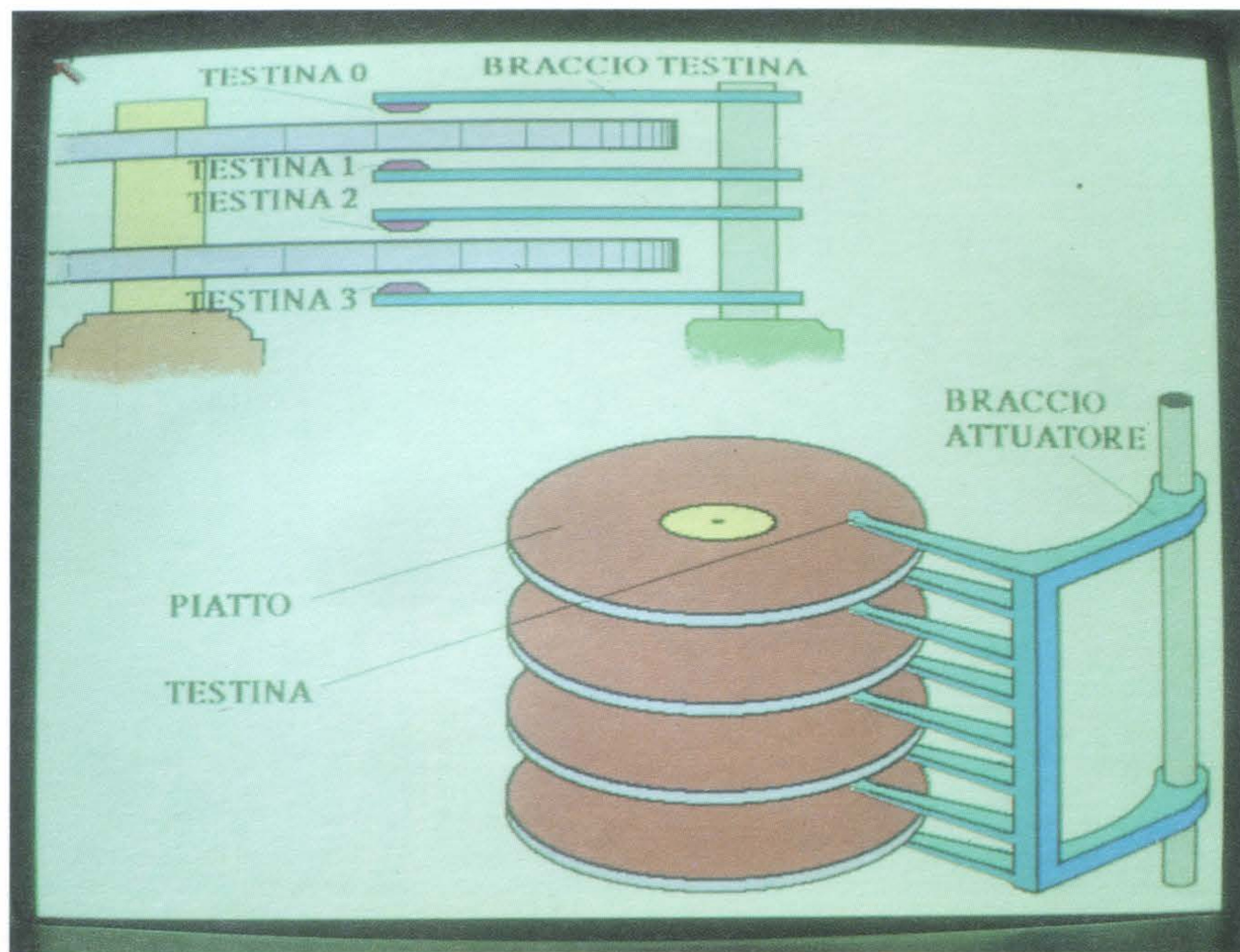



Fig. 1. Un hard disk è in realtà costituito da più dischi sovrapposti, ad ognuno dei quali corrispondono una o più testine magnetiche.

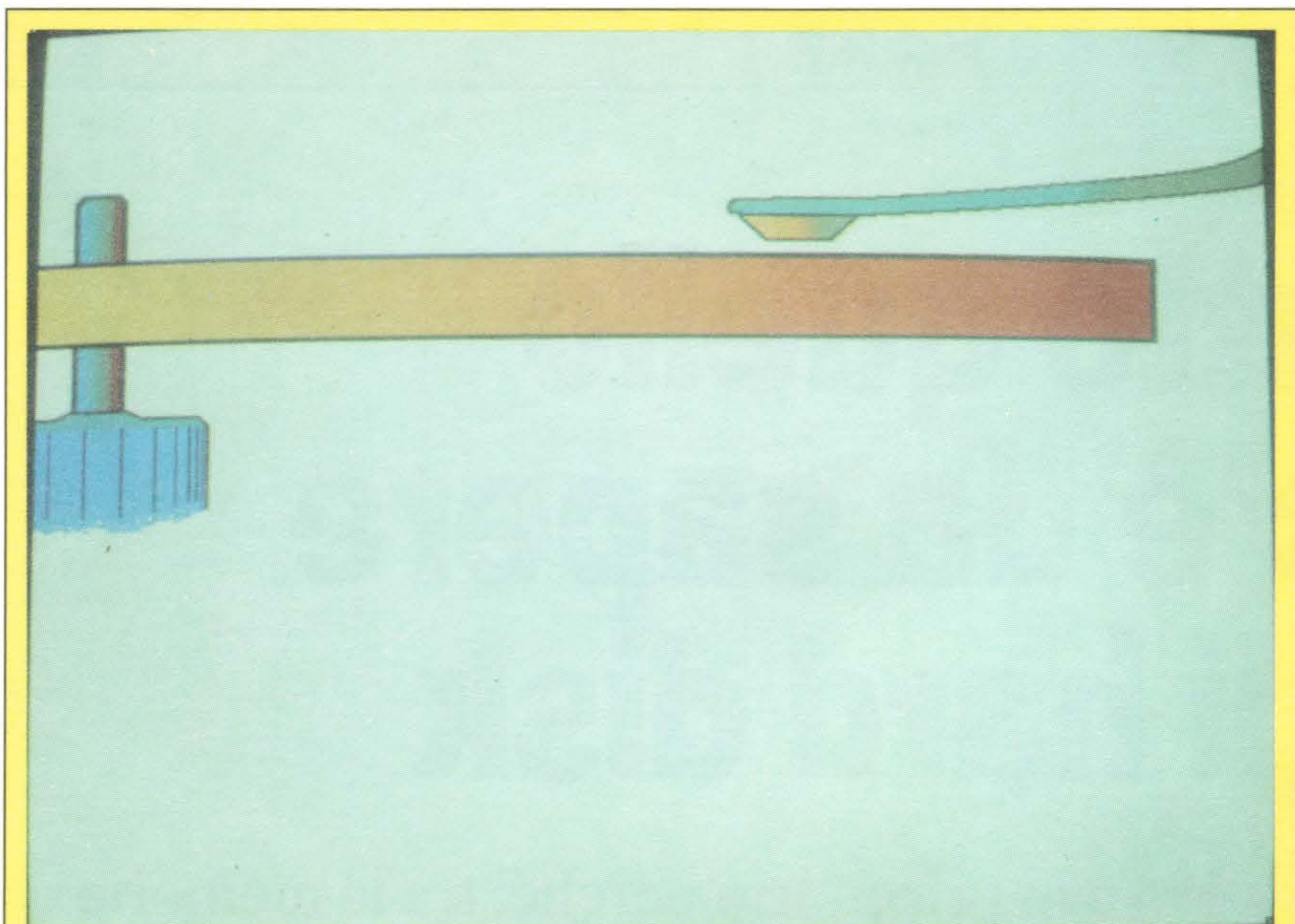



Fig. 2. La testina non sfiora la superficie del disco durante gli accessi al disco, senza venire mai a contatto diretto con la superficie, tranne che in apposite zone di parcheggio.

magnetico, fissati ad un asse centrale in rotazione. Ad ogni disco sono associate una o più testine magnetiche libere di muoversi orizzontalmente dalla periferia del disco verso il suo centro (Fig. 1).

La combinazione di movimenti dei dischi e delle testine consente a queste ultime di raggiungere qualsiasi punto sulla superificie dei dischi.

L'evoluzione della periferica ha portato ad una progressiva riduzione della grandezza dei dischi (e quindi della loro superficie) insieme ad un aumento della densità di dati memorizzabili (parametro indicato come **tpi** dall'inglese: *tracks per inch*, tracce per pollice).

Le testine non toccano mai la superficie dei dischi, ma vi volano letteralmente sopra, sospese da un cuscino d'aria creato dalla rapida rotazione, ad una quota di 2 millesimi di millimetro. In proporzione, la testina può essere paragonata ad un Jumbo che, a 3 cm. di altezza dal suolo, percorra la propria traiettoria ad una velocità pari a MACH 4, equivalenti a circa 4800 Km/h!

Ogni testina, il cui movimento viene controllato da un motore detto attuatore, si comporta come un aliante, capace di decollare, planare ed atterrare! (Fig. 2).

IL PARCHEGGIO DELLE TESTINE

Durante i periodi di inattività o di non utilizzo (ad esempio allo spegnimento del calcolatore), le testine vengono parcheggiate in posizione sicura, lontano dalle aree contenenti dati. La maggior parte dei dischi rigidi è dotata di **autoparking**, vale a dire della capacità automatica di parcheggio delle testine. Le testine vanno assolutamente parcheggiate durante la manutenzione ed il trasporto di un disco rigido per evitarne il danneggiamento. Dal momento che la superficie magnetica non viene toccata, i dischi non vanno teoricamente incontro ad usura e costringono d'altra parte i costruttori a realizzare i supporti in un'atmosfera assolutamente priva di polvere o di altre particelle, ed a sigillare infine il tutto all'interno di un contenitore dotato di sospensioni anti urto ed anti-vibrazioni.

Nonostante le severe norme adottate durante la costruzione, è rarissimo che i dischi non presentino imperfezioni (seppur ridotte al minimo); un controllo in fabbrica evidenzia gli eventuali difetti ed esclude automaticamente il passaggio delle testine nelle zone difettose (in genere indicate da un'etichetta all'esterno dell'involucro).

Tutte le norme di costruzione esaminate sono responsabili della maggiore velocità di lettura e scrittura, paragonata a quella dei floppy disk (un buon disco rigido trasferisce una quantità di dati pari al contenuto di un dischetto in circa *un secondo*), oltre che della grande capacità di immagazzinamento (si parte di solito da 20 per arrivare fino a 600 Megabyte ed oltre). Un hard disk necessita, come è facile immaginare, di una meccanica di funzionamento molto precisa ed affidabile.

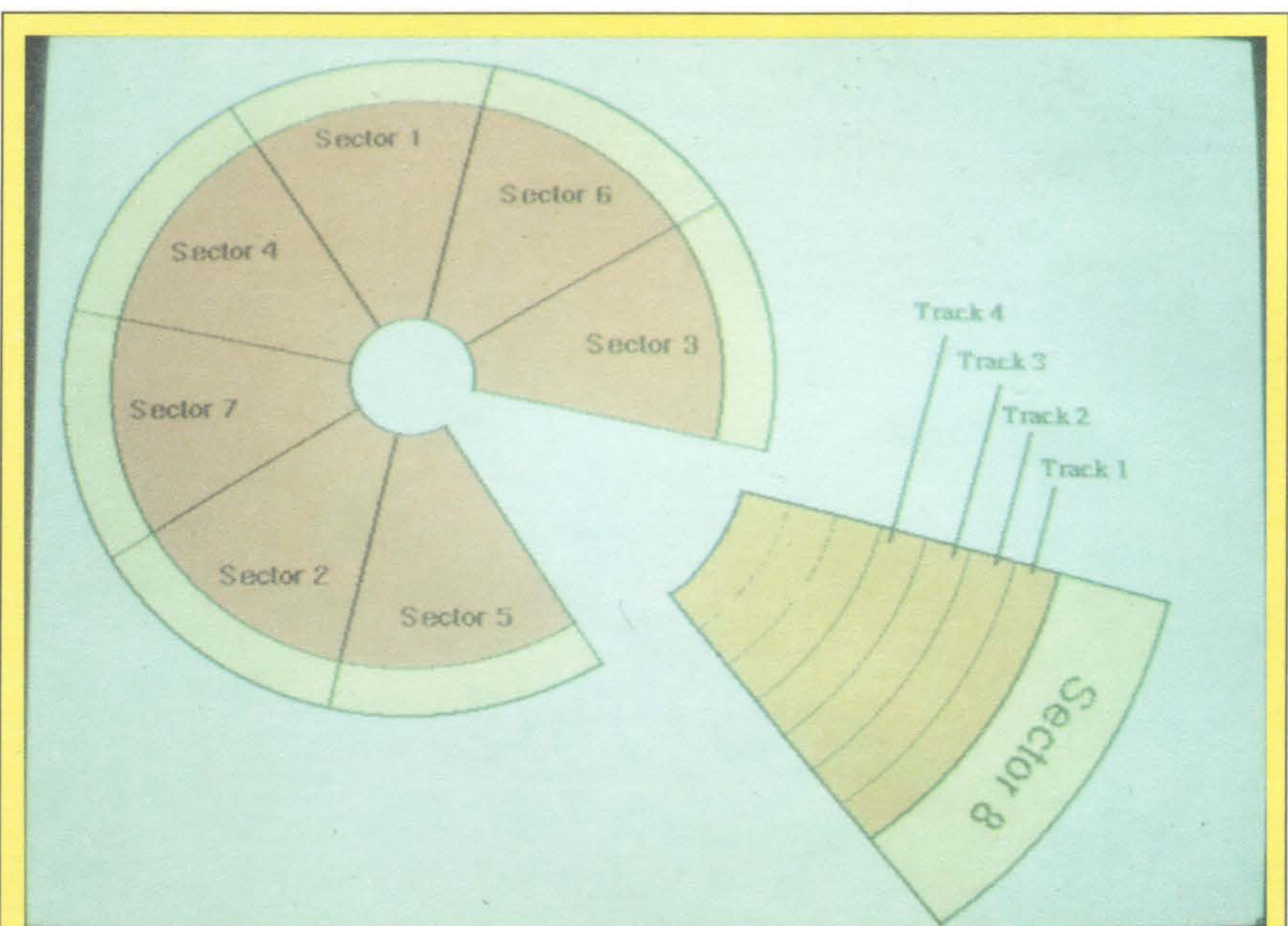



Fig. 3. Ogni disco è suddiviso in tracce concentriche, ognuna divisa in vari settori. Il fattore di interleave determina la sequenza esatta nella quale i settori vengono letti o scritti.

IL FORMATO DEI DATI

Il sistema operativo organizza i dati così come fa per un floppy. Ciascun disco viene suddiviso in porzioni circolari concentriche chiamate **tracce**, e ciascuna di queste tracce in porzioni chiamate **settori** (Fig. 3). Ogni settore può contenere 512 byte di dati. Questa organizzazione (utilizzata dal sistema operativo per il corretto reperimento dei dati) viene effettuata al momento della formattazione e vale per ognuna delle due facce costituenti ciascun disco.

Per la maggior parte degli utenti concetti come tracce, settori o cilindri sono di nessuna utilità dal momento che l'AmigaDos s'incarica di gestire il tutto in maniera trasparente. In alcune condizioni però, conoscerne i meccanismi consente di capire ciò che può essere successo al proprio disco rigido (errori di scrittura o lettura, infezioni «virali», etc.).

Concetto basilare è invece quello di **partizione**. Una partizione consiste in una suddivisione software del proprio hard disk. In genere è possibile suddividere un hard disk in innumerevoli partizioni, ciascuna contrassegnata da un nome (solitamente **DH0:**, **DH1:**, **DH2:**, oppure con una denominazione che ne indichi il contenuto, come **Work:**, **Music:**, **Paint:** etc.). La creazione di partizioni è un'operazione utile, soprattutto per hard disk di grossa capacità, ai fini di un'intelligente suddivisione del loro contenuto, come avremo modo di approfondire in seguito. È possibile inoltre riservare partizioni a sistemi operativi diversi, quali **MS-Dos** e **Macintosh**. Questo significa che porzioni del disco rigido sono riorganizzabili secondo le specifiche di altri sistemi operativi, rendendo possibile l'uso di quei pacchetti software progettati appositamente per tali sistemi e che richiedono obbligatoriamente la presenza di un hard disk.

LA SCELTA DI UN DISCO RIGIDO

Esistono numerosi hard disk per Amiga aventi differenti capacità e caratteristiche. La scelta di un disco rigido dipende dal modello di Amiga posseduto (Amiga 500, 1000 o 2000: il modello 3000 viene venduto già con un hard disk installato), da quanto s'intende investire, e dalle proprie esigenze operative. In genere il parametro del quale tener più conto è il costo rapportato alla capacità, alla velocità d'accesso, ed al tipo di controller.

COMANDI DELL'AMIGADOS UTILI PER LA GESTIONE DEL DISCO RIGIDO:

- LOCK protegge una partzione da scrittura.
- CD cambio directory.
- DIR elenco directory.
- LIST elenco attributi file.
- FORMAT cancellazione e preparazione.
- MAKEDIR crea nuova directory.
- DELETE cancella file.
- RELABEL cambia il nome al disco.
- RENAME cambia nome al file.
- COPY copia del file
- PROTECT cambia bit di conrollo file.
- FILENOTE aggiunge commenti a file.

Fig. 4. I comandi AmigaDos usati nella gestione del disco fisso sono più o meno gli stessi impiegati con i floppy disk.

Quest'ultima parte merita un discorso più approfondito. Ogni hard disk infatti è dotato di un dispositivo **Controller**, una parte hardware cioè capace di interfacciare la periferica al sistema operativo mediante l'invio di comandi specifici, e responsabile del trasferimento dati. I controller più diffusi sono i cosiddetti **ST-506** e **SCSI** (pronuncia «scasi»).

I controller di tipo **ST-506** consentono di collegare hard disk del tipo omonimo, molto diffusi sui computer IBM e compatibili. Sono in genere i più vecchi, i meno costosi e veloci.

Quelli di tipo **SCSI** (Small Computer Standard Interface) sono stati adottati dalla Apple e da diverse altre case (compresa la Commodore) perché molto più vantaggiosi. Ad un singolo controller SCSI possono essere facilmente connesse fino ad otto periferiche (hard disk, unità a nastro, dischi ottici) variamente assortite, o addirittura altri controller (in questo caso, i dispositivi vengono condivisi).

MOUNTLIST TIPO PER UNA SINGOLA PARTIZIONE

```
DH0:
    Device=gvpscsi.device
    FileSystem=L:FastFileSystem
    Unit=000
    Flags=0
    Surfaces=4
    BlocksPerTrack=33
    Reserved=2
    Interleave=0
    LowCyl=1
    HighCyl=670
    Buffers=32
    GlobVec=-1
    BufMemType=1
    Mask=0xFFFFFE
    MaxTransfer=0x7FFFFFFF
    Mount=1
    DosType=0x444f5301
    StackSize=4000
#
```

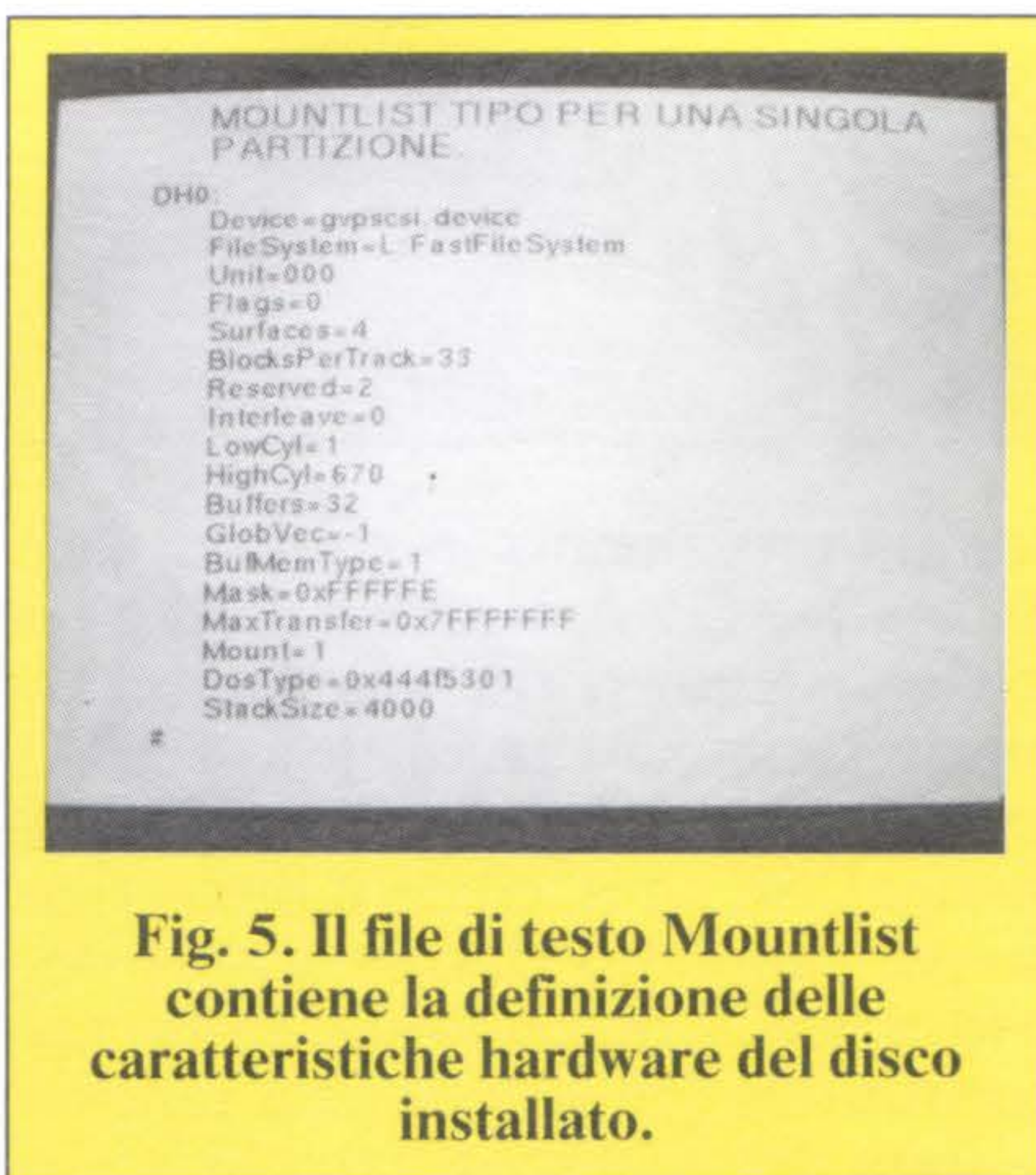



Fig. 5. Il file di testo Mountlist contiene la definizione delle caratteristiche hardware del disco installato.

Il mercato offre un numero considerevole di controller e di hard disk, venduti sia in coppia che separatamente. Inoltre molte aziende (GVP, Supra, ICD, Nexus) forniscono soluzioni con schede di memoria incorporate e/o acceleratori. L'analisi dei vari modelli esula dagli intenti di questa esposizione; rimandiamo pertanto agli articoli che AmigaByte ha pubblicato e che continuerà a dedicare a ciascun prodotto.

MONTARE UN HARD DISK

Il montaggio di un hard disk è molto semplice e spesso si limita all'inserimento di una scheda con relativo collegamento di un led di funzionamento e di un cavo di connessione, oppure al posizionamento di un'unità esterna (di solito quest'ultima alternativa è appannaggio dei modelli Amiga 500). Generalmente ogni costruttore fornisce, insieme ai propri modelli, un programma adatto, in grado di controllare il corretto funzionamento dell'hardware, di realizzare la formattazione dell'hard disk e la sua suddivisione in una o più partizioni (operazione detta di **Setup**). Operazioni che comunque sono eseguibili, pur se più macchinosamente, anche da CLI tramite i comandi riportati in Fig. 4.

L'hard disk viene in genere fornito già formattato, a meno di non acquistarlo separatamente dal controller.

A partire dalla versione 1.3 del Kickstart, Amiga è capace di eseguire l'**Autoboot** da una partizione hard disk. Questo significa che il computer può caricare automaticamente il WorkBench e tutti i file assegnati all'accensione del calcolatore leggendoli da una partizione del disco rigido, senza che sia presente un disco di sistema nel drive interno. Inoltre, l'implementazione del Fast File System (anch'essa risalente al WorkBench 1.3) aumenta la velocità d'accesso e di utilizzo dei file presenti sul disco rigido, oltre che aumentarne la capacità di memorizzazione: si ha un guadagno di circa 50 Kbyte per ogni Megabyte.

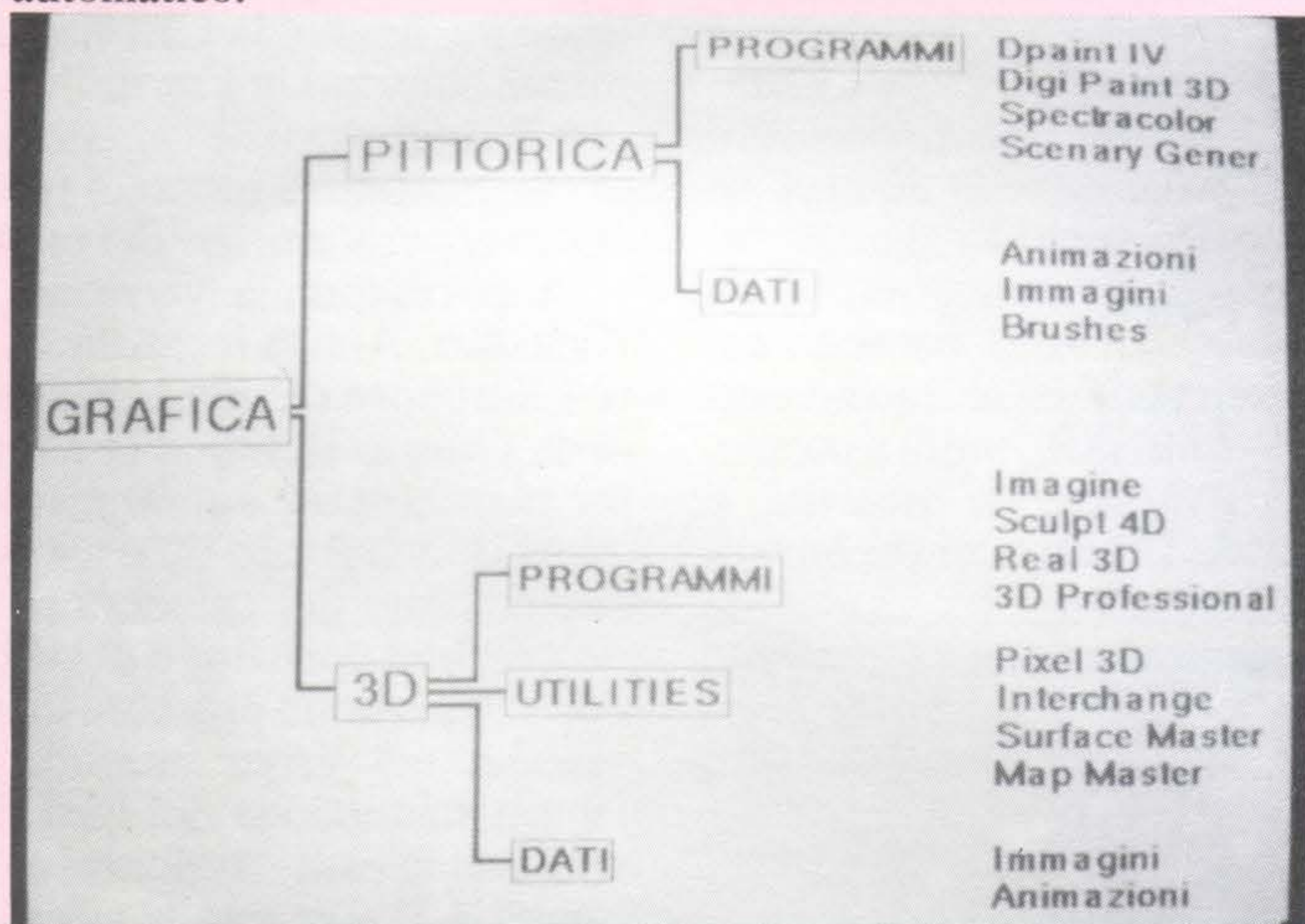
LA MOUNTLIST

Non tutti i controller (e non tutte le partizioni) sono autoconfiguranti,

ORGANIZZAZIONE E MANUTENZIONE

La suddivisione in due o più partizioni torna utile quando si dispone di un hard disk di grossa capacità e si desidera ordinare il contenuto in sezioni separate. Ciò consente senz'altro di organizzare meglio programmi e dati, oltre che velocizzare e migliorare la manutenzione. Per esempio se si desidera utilizzare un hard disk per lo più per applicazioni grafiche si possono creare due partizioni denominate

DH0: WORKBENCH
DH1: GRAFICA

Nella prima, dalla quale eseguiremo il boot, sistemiamo i programmi di sistema, di manutenzione o comunque quelli non strettamente attinenti le applicazioni grafiche per le quali abbiamo riservato la seconda partizione.
Naturalmente niente ci vieta di suddividere il disco rigido in un maggior numero di partizioni: abbiamo visto come questo dipenda dalle necessità personali e dall'uso intenzionale. Ciò che deve risultare chiaro è che già una suddivisione in più partizioni aiuta ad organizzare meglio lo spazio a disposizione. Ricordate però che ad ogni partizione aggiunta corrisponde un programma di gestione, e quindi una diminuzione (seppur leggera) della velocità della macchina.
Sappiamo che AmigaDos, come tutti i sistemi operativi degni di questo nome, permette di costruire directory e sotto-directory. È proprio avvalendoci di questa capacità che possiamo introdurre un ulteriore livello organizzativo.
Sistemando programmi e dati in una struttura ramificata ad albero, questi risultano di facile reperimento e velocemente si può provvedere al loro aggiornamento. Supponendo di suddividere la nostra seconda partizione contenente materiale grafico, possiamo ipotizzare un'organizzazione ideale mostrata in figura.
Effettuata quest'opera di suddivisione (a seconda delle necessità di ognuno, e comunque è buona norma eseguirla anche grossolanamente, ed è sempre possibile affinare successivamente l'organizzazione aggiungendo o spostando directory, sotto directory e relativo contenuto), possiamo finalmente iniziare la copia dei programmi da floppy a hard-disk.
È bene dire immediatamente che non esiste una procedura universale per la copia, ma che devono essere seguite le istruzioni fornite di volta in volta dalla casa produttrice del software in questione. Molto spesso i dischetti comprendono un programma denominato Install o HDInstall, che si occupa del trasferimento in modo automatico.

cioè capaci di avvertire automaticamente il sistema della loro presenza e delle loro modalità di funzionamento. Subito dopo il boot, dunque, (che deve sempre avvenire da floppy o da una partizione autoconfigurante dell'hard disk), può essere necessaria una procedura di identificazione.

I controller vengono solitamente forniti con un **device driver**, ovvero con un programma in grado di fare da tramite fra AmigaDos ed il controller. Alcune periferiche (ad esempio le stampanti parallele o seriali) hanno già un device all'interno di AmigaDos; altre, come il disco rigido per l'appunto, devono essere dotate di device driver forniti dalla casa costruttrice (un'eccezione è rappresentata dal Kickstart 2.0, che comprende lo **scsi.device**). A setup avvenuto, in genere è possibile trovare il driver nella **directory DEVS:** oppure **Expansion**. Il nome del drive è importante perché riportato in un file di testo speciale definito **Mountlist** e posizionato solitamente nella **directory DEVS**.

Questo file contiene caratteristiche particolareggiate circa l'hardware montato. Vale la pena di esaminarlo in dettaglio, prendendo come esempio quello riportato in fig. 5.

DH0: è il nome assegnato dal sistema alla partizione, e può essere cambiato a piacere. Di solito è utile attribuire nomi descrittivi del contenuto della partizione. Se si opta per più partizioni, è conveniente non superare i tre caratteri di lunghezza dal momento che il nome verrà riportato poi nei file requester dei vari programmi, e non tutti sanno gestire correttamente periferiche dal nome troppo lungo.

Il nome della partizione è seguito dal nome del device driver contenuto, come abbiamo detto, nella directory Devs. Le linee che seguono sono composte da una parola chiave seguita da un «=» e da un valore: si tratta di parametri letti dal sistema operativo ai quali sono assegnati i valori stituati dopo il segno di uguale.

Unit specifica il numero di unità. Si tenga conto che l'hardware (della GVP) relativo alla mountlist dimostrativa è di tipo SCSI per cui, come abbiamo detto, al controller possono essere agganciate fino ad otto periferiche della stessa natura. Il controller individua ciascuna unità assegnandole un numero.

Il valore di **Flags** viene passato tale e quale al device driver; il suo significato viene definito dal programmatore, e varia da un modello all'altro. Le due linee che seguono contengono informazioni sull'organizzazione del disco rigido.

Surfaces specifica il numero di testine: in genere il programma di setup è capace di rilevare automaticamente il numero di testine presenti, oppure i costruttori forniscono elenchi delle caratteristiche proprie di un disco rigido (se si cambia hard disk con diverse caratteristiche questo parametro va modificato).

SEGUE PAG 42

TOOLS

a cura di Guglielmo Cancelli

M.EDIT ACTION 1.1

Fra gli appassionati di musica elettronica, molti conosceranno la famosa workstation musicale **M1** della **Korg**, la sua versione expander (**M1R**) od il più economico, ma non per questo meno potente, expander **M3R**. I timbri, caldi e pastosi, ed il notevole numero di effetti digitali di cui sono forniti questi sintetizzatori, hanno fatto breccia in tanti musicisti alla ricerca di sonorità diverse da quelle offerte dalle workstation in sintesi FM.

Tutti coloro che hanno deciso di creare ex-novo un programma dalle caratteristiche particolari, o di realizzare un kit di percussioni personalizzato, si sono dovuti presto scontrare con la difficoltosa editazione dei parametri imposta dal limitato display di cui sono dotate queste macchine.

Oggi i musicisti possono contare su «**M.edit Action**», un potente editor, completamente in lingua italiana, che unisce ad un ambiente di lavoro veramente user friendly la comodità del multitasking ed una veste grafica particolarmente curata.

Il programma è strutturato in quattro pagine destinate a gestire le memorie di suono (**programs**), le memorie di performance (**combinations**), i kit di percussione, ed i dati **global**.

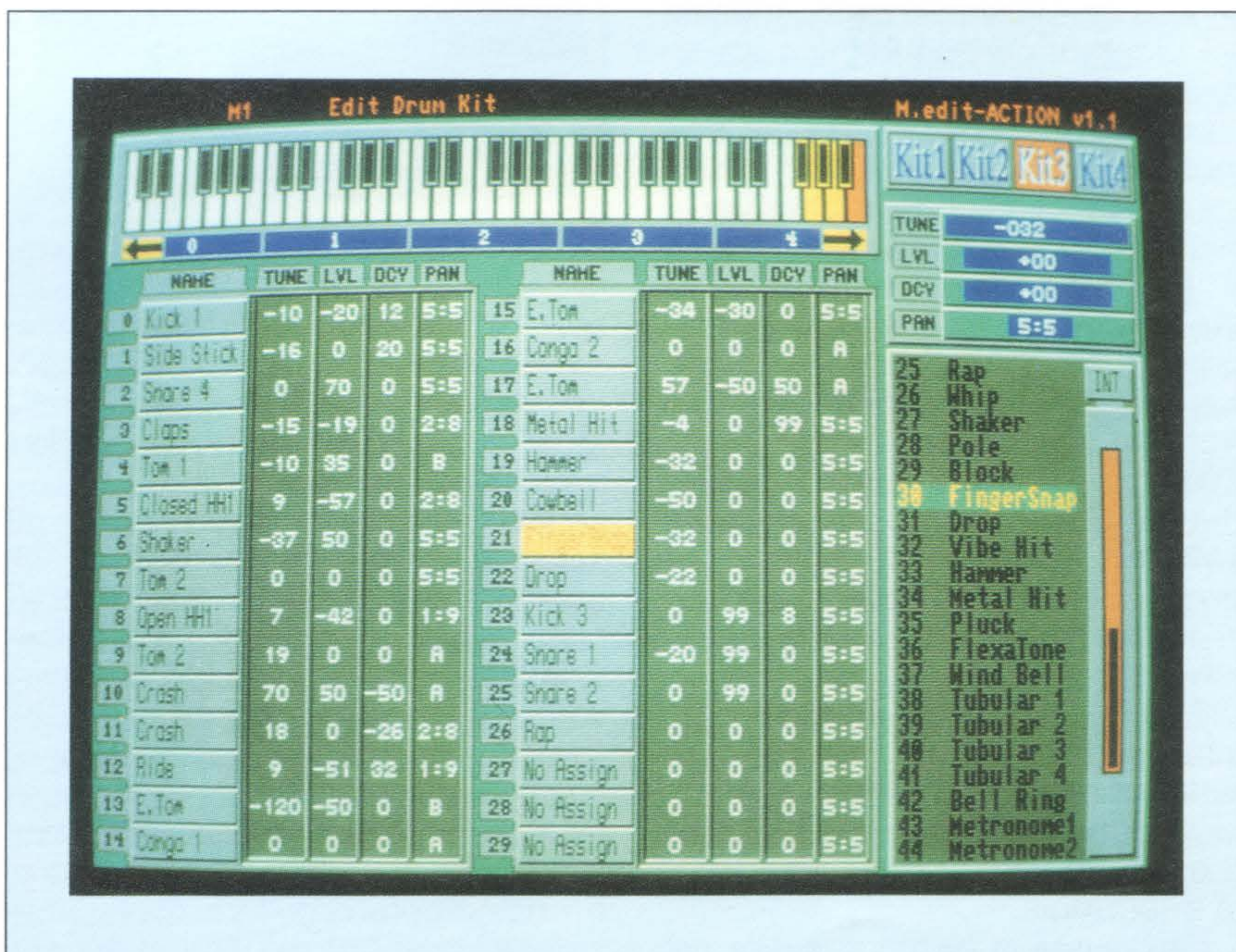
Dopo il lancio, il programma controlla la presenza del collegamento con il sintetizzatore. Se il collegamento risulta presente, esegue il dump di tutti i dati esistenti nella memoria interna dell'M1; in caso contrario, carica da disco i dati di default.

In questa fase si suppone che il canale comune (**COMMON CH**) sia impostato ad 1 sul synth. Questa assunzione non deve essere vista come una limitazione poiché è possibile avere necessità di gestire più di un M1 contemporaneamente, ed ogni sintetizzatore deve essere univocamente distinto. Il canale comune assolve a questo compito, cosicché si può, memoria permettendo, lanciare fino a 16 volte «M.edit Action», così da poter gestire un equivalente numero di sintetizzatori.

La prima pagina che automaticamente si apre è quella relativa al modo **Edit Program**. La zona superiore mostra delle informazioni che risulteranno sempre presenti in questa modalità.

Clickando sul nome del *program* corrente si ottiene la possibilità di rinominarlo. Tutti i gadget presenti in questa zona sono abbastanza autoesplicativi e l'unico appunto da fare riguarda la triade di gadget a forma di frecce, il cui scopo consiste nello spostamento all'interno degli elementi ai quali i gadget si riferiscono.

La zona inferiore è destinata a diversi usi: potrà visualizzare e quindi rendere possibile la selezione diretta di un elemento (**program** o **multisound**) fra quelli appartenenti allo stesso gruppo (o allo stesso banco, nel caso dei programs), oppure riporterà i parametri che si riferiscono alla sezione scelta dal menu **Edit1**.

Le sezioni alle quali si può accedere tramite questo menu riguardano i dati relativi al **Pitch**, ovvero il modo in cui varierà la frequenza del suono durante il tempo in cui un tasto risulta abbassato e quindi sollevato; al **VDF** (Filtro Variabile Digitale), che controlla la qualità timbrica del suono tagliando le alte frequenze che compongono il *multisound*; (Amplificatore Variabile Digitale), che agisce sul volume della forma d'onda del suono; al **Controller**, che si fa carico di gestire quattro specifici gruppi di parametri (pitch modulation, vdf modulation, after touch e joystick); ed infine all'**Effect**, che fornisce tutte le informazioni

sulle due unità digitali multieffetto.

Queste unità sono stereofoniche, paragonabili a dispositivi stand alone, in grado di fornire una vasta gamma di effetti (Chorus, Leslie, Distorsore, Flanger, Riverbero, Phaser, etc). In questa pagina è possibile entrare in modo Operazioni per effettuare scambi e movimenti, inizializzazioni o comparazioni fra diversi *program*.

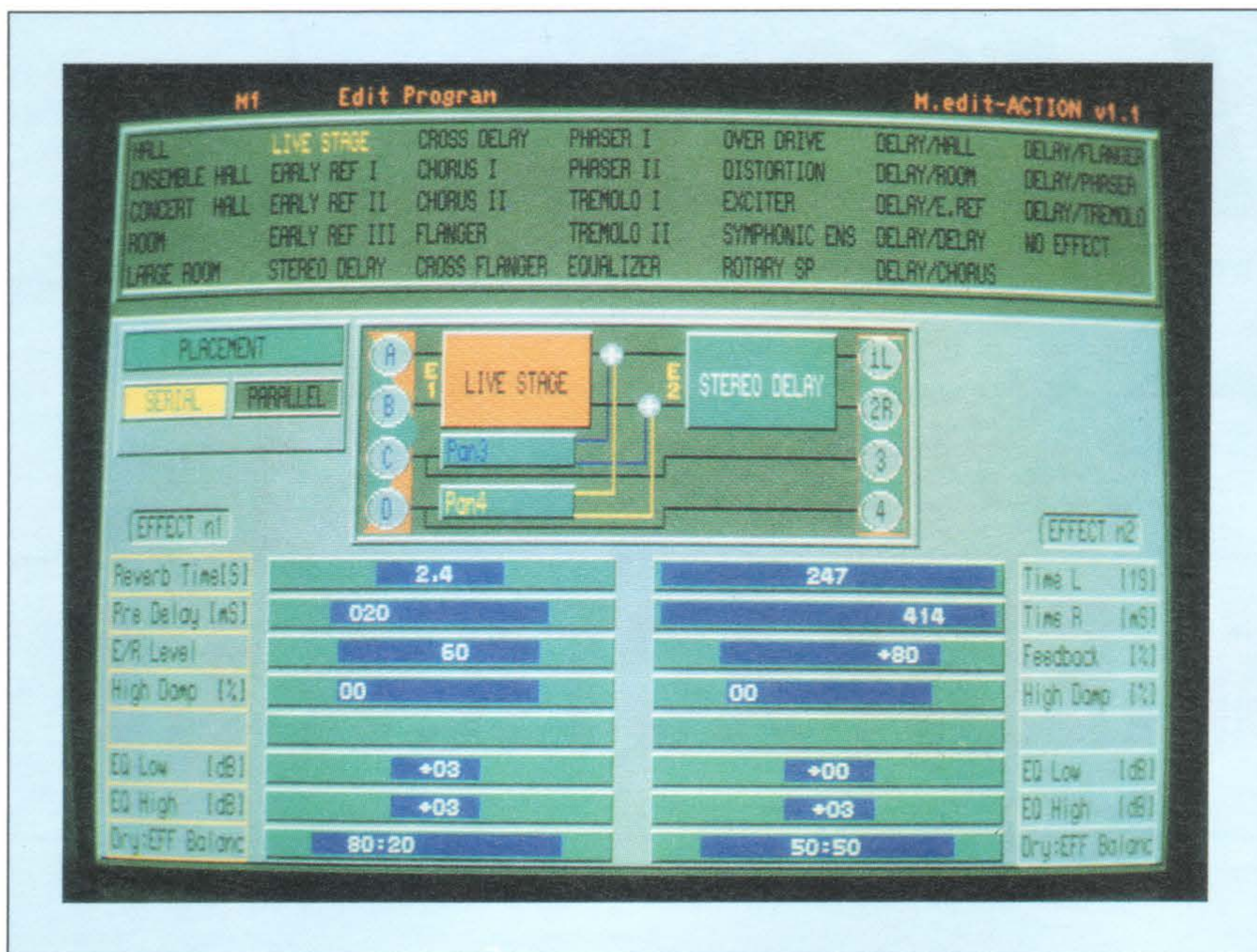
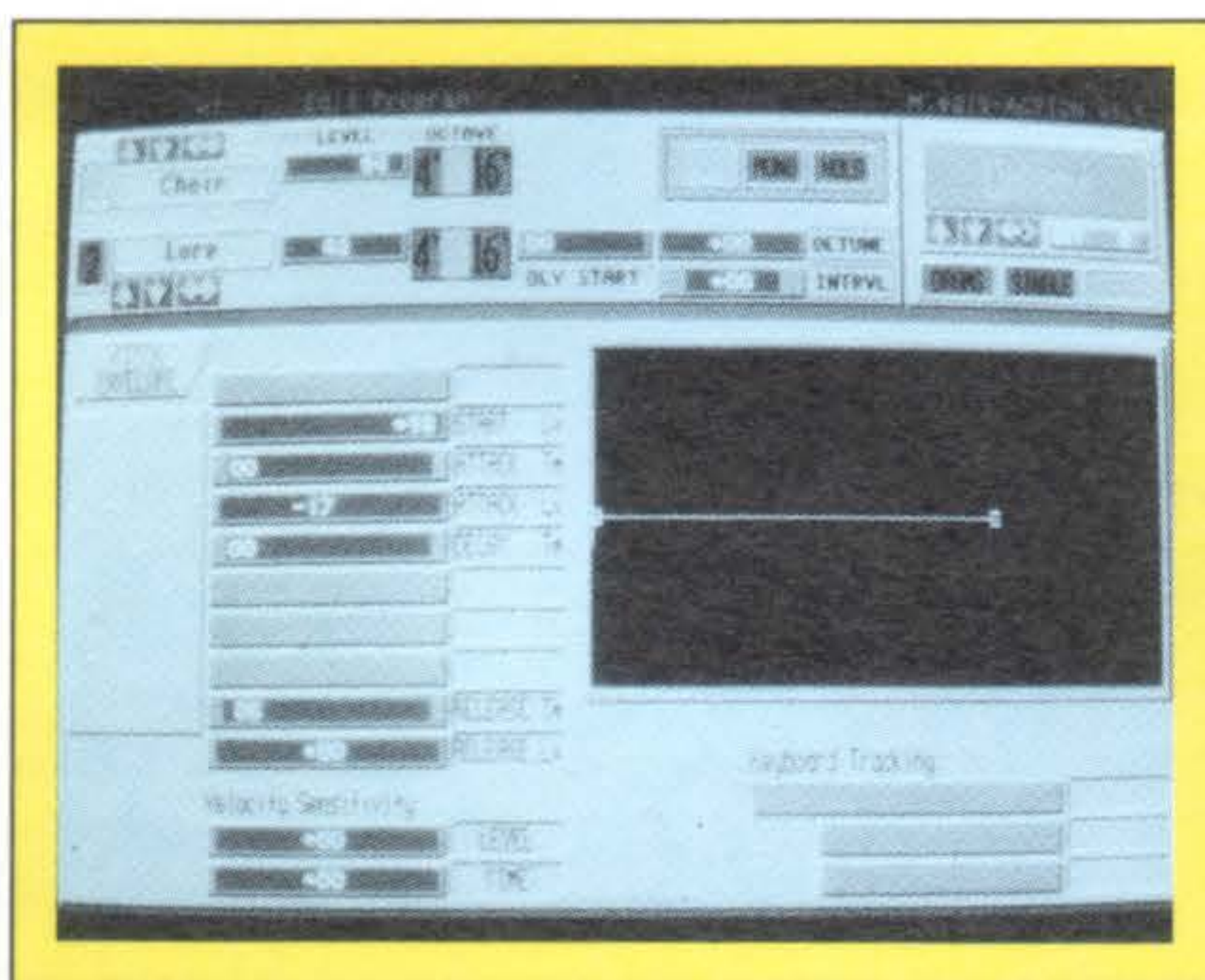
Di notevole interesse è il comando **Compara**: clickando su questo riquadro, quando entrambi i *program* sono dello stesso tipo (double, single o drum), si accede ad una modalità in cui è possibile comparare, e quindi copiare, anche un singolo parametro dal program sorgente a quello destinazione.

I campi circondati da un rettangolo giallo indicano che il valore di quei parametri risulta differente negli omonimi del program destinazione. Un click del mouse su uno di essi invoca l'acquisizione del valore contenuto nel medesimo campo appartenente al program sorgente. L'operazione è resa visibile dal rettangolo rosso che rimpiazza quello giallo su cui si è clickato. Clickando nuovamente sul campo viene ripristinato il valore di partenza che apparteneva al program destinazione. Un ultimo click sul gadget **Copia** rende definitive le modifiche apportate al program destinazione.

Nel caso in cui si sia selezionato lo stesso program sia come sorgente che come destinazione, sarà disponibile un gadget che permetterà di stampare i valori dei parametri del program con un formato identico a quello che si otterrebbe visualizzando gli stessi nel display presente sul sintetizzatore.

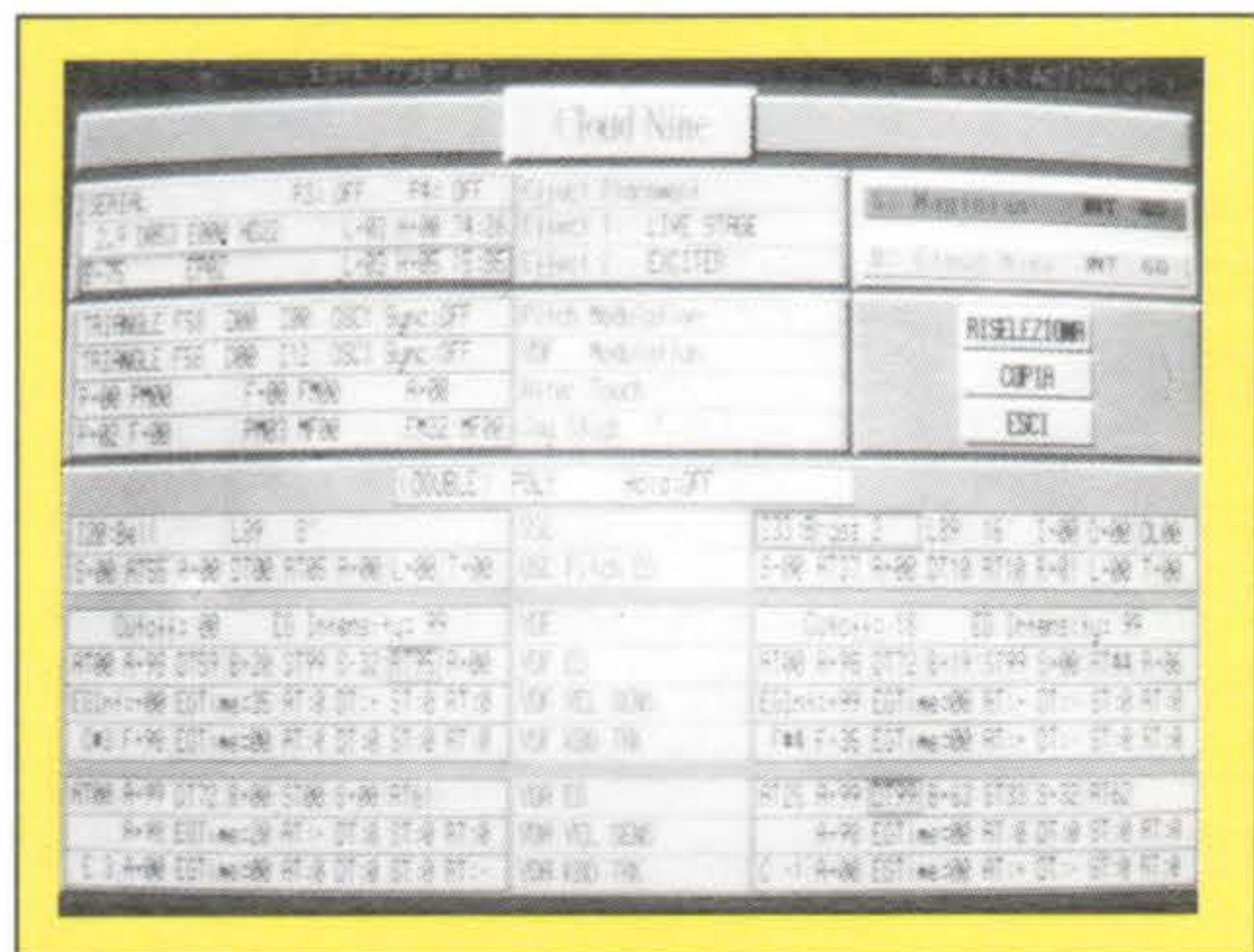
La pagina **Edit Combination** è dedicata all'editing delle memorie di **performance** o combinations, che permettono di combinare i program, di cui sono composte, in cinque diversi modi: **Single** (un solo program); **Split** (due program in due diverse zone della tastiera); **Layer** (due program sovrapposti); **Velocity Switch** (due program in due diverse aree di dinamica); **Multi** (fino ad un massimo di otto program assegnabili come si desidera).

Il tipo di combination più sfruttato sarà ovviamente Multi, che si adatta perfettamente all'impiego con un sequencer, per offrire l'opportunità di eseguire fino ad otto linee diverse di polifonia.

Da rilevare la funzione svolta dal gadget **Full List**: clickando su di esso si ottiene, nell'area inferiore, la visualizzazione di tutti i parametri che appartengono ai program che formano il tipo Multi. Anche in questa pagina è possibile entrare in modo Operazioni per utilizzare i comandi visti in precedenza, ma agendo in questo caso sulle combination.

Tutti coloro che hanno cercato di rifare un Drum Kit utilizzando il display LCD a 40 caratteri presente su M1 avranno speso molto tempo per raggiungere i risultati sperati. Molti altri, forse, avranno abbandonato l'impresa considerando tedioso il sistema fornito dalla Korg, e si saranno dovuti accontentare di usufruire dei tre kit di default presenti sul sintetizzatore. La pagina **Edit Drum Kit** ha lo scopo di eliminare ogni possibile dubbio circa la creazione/modifica di un kit di percussione personalizzato. Con pochi click del mouse è possibile risparmiarsi una notevole mole di lavoro e contemporaneamente avere una visione chiara di ciò che si sta facendo.

La tastiera mostrata in alto possiede tre modalità di funzionamento: quando nessun percussivo è selezionato, essa visualizza (in rosso), in un range di cinque ottave, tutti i tasti che possiedono un'assegnazione diretta di un percussivo. Per sapere quale specifico percussivo appartiene ad un tasto, basta clickare sul tasto stesso; così facendo si ottengono (in giallo) anche i tasti che indirettamente rispondono al percussivo del tasto fondamentale (tasto rosso).

Ovviamente è anche possibile reindirizzare un percussivo verso uno dei tasti che fino a quel momento non è stato assegnato in maniera diretta. In parole povere, non si può chiedere ad un tasto di fornire più di una percussione.

La pagina **Edit Global** presiede alla gestione dei parametri globali del synth. Risulta la meno complessa fra tutte quelle menzionate precedentemente e possiede importanti comandi, come la ricanalizzazione automatica. Questo comando

sarà di notevole aiuto quando si avrà una combination di tipo Multi all'interno della quale i program che la compongono risultano assegnati ad un canale diverso da quello comune.

«M.edit Action», del quale sono disponibili due versioni (una per M1/M1R ed una riservata a M3R), è un programma decisamente interessante e versatile, anche se il suo pubblico è ovviamente limitato ai possessori di un synth Korg. Degno di nota è anche il fatto che comprenda la documentazione nella nostra lingua, essendo stato sviluppato da un programmatore italiano.

Distribuito da:
DIGITAL MUSIC SERVICE
Via Ugo La Malfa 1
20068 Peschiera Borromeo (MI)

SUPERBASE PROFESSIONAL 4

Tra i programmi talmente noti da non avere bisogno di presentazioni c'è sicuramente «**SuperBase**», il database della Precision Software originariamente apparso anni fa sugli schermi del Commodore 64 e convertito poi su di una serie di piattaforme diverse, tra le quali i sistemi Atari, MsDos ed Amiga.

Le versioni precedenti di questo database relazionale, fino alla release 3, avevano goduto di grande popolarità tra gli utenti Amiga, ed in particolare tra quelli italiani, grazie all'edizione tradotta nella nostra lingua alcuni anni orsono dal distributore CTO (recensita su AmigaByte n. 6).

Lo sviluppo di «SuperBase» per Amiga si era poi apparentemente fermato a quella versione, mentre su altri sistemi sono uscite nuove release aggiornate del programma.

È stato il successo riscosso da «SuperBase 4» per **Windows**, su PC Ibm e compatibili, che ha probabilmente spinto la Precision Software ad aggiornare il suo più noto programma per produrlo anche per Amiga. Tutte le versioni di «SuperBase» per Amiga sono disponibili solitamente sotto due diverse vesti: la variante «**Personal**», priva della programmabilità e dalle prestazioni ridotte, è rivolta all'utenza hobbistica, mentre quella «**Professional**», più potente e completa, richiede maggiore capacità di memoria.

Per il momento di «SuperBase 4» è disponibile solo quest'ultima versione, che in teoria richiede un minimo di 1,5 mega di memoria e due drive, ma che può in realtà essere sfruttata efficacemente soltanto su Amiga dotati di hard disk.

Normalmente non ci soffermiamo troppo sulla descrizione della confezione dei programmi, ma nel caso di «SuperBase Professional 4» il «packaging» è troppo impressionante per non meritare una citazione: la confezione contiene tre manuali (uno per il database vero e proprio, un altro per il Forms Designer e per il linguaggio DML, ed un terzo ricco di esempi e dimostrazioni d'uso pratico del programma) per un totale di circa 900 pagine, più i due dischetti del programma.

Le precedenti versioni di «SuperBase» sfruttavano, come metodo di protezione dalla copia, l'odiosissimo **Dongle**: una chiave hardware che doveva essere inserita nella porta joystick di Amiga, senza la quale il programma si rifiutava di funzionare. La protezione è stata abbandonata a favore di una semplice procedura di installazione serializzata: la prima volta che il programma viene usato, occorre digitare un numero di serie (incluso nella confezione) ed i propri dati anagrafici, che verranno permanentemente memorizzati all'interno di «SuperBase». Il programma installato può essere poi tranquillamente copiato (per mantenere un backup) senza problemi.

Nei confronti delle versioni precedenti, «SuperBase» mantiene la compatibilità nel formato dei dati; la capacità di memorizzare e di ricerca è stata ora portata all'incredibile cifra di un miliardo (!) di record. Oltre che dei dati alfanumerici, ogni record può contenere campi rappresentati da immagini grafiche o da suoni campionati. La struttura dei record è stata modificata con l'aggiunta di un nuovo tipo di campo logico (che può assumere i valori vero/falso).

Grazie alla sua doppia natura di programma per Amiga e per Win-

dows, «SuperBase Pro 4» supporta molte caratteristiche solitamente disponibili solo su programmi di archiviazione sviluppati per Pc Ibm e compatibili: ad esempio può importare dati generati da alcuni tra i più noti fogli elettronici per sistemi MsDos, tra i quali «Lotus 123», «Excel» e «Visicalc», oltre che da «Logistix» e «SuperPlan» della stessa Precision Software, disponibili anche su Amiga. Anche gli archivi in formato .DBF, creati con il celeberrimo «**dBase III**», possono essere importati da «SuperBase»: sfortunatamente i dati possono essere letti e scritti nello stesso formato ma non modificati direttamente. L'interscambio di dati tra sistemi diversi è comunque facilitato dalla possibilità di trasmettere dati direttamente attraverso la porta seriale (o attraverso un modem ad essa collegato), trasferendoli con il protocollo **X-Modem**.

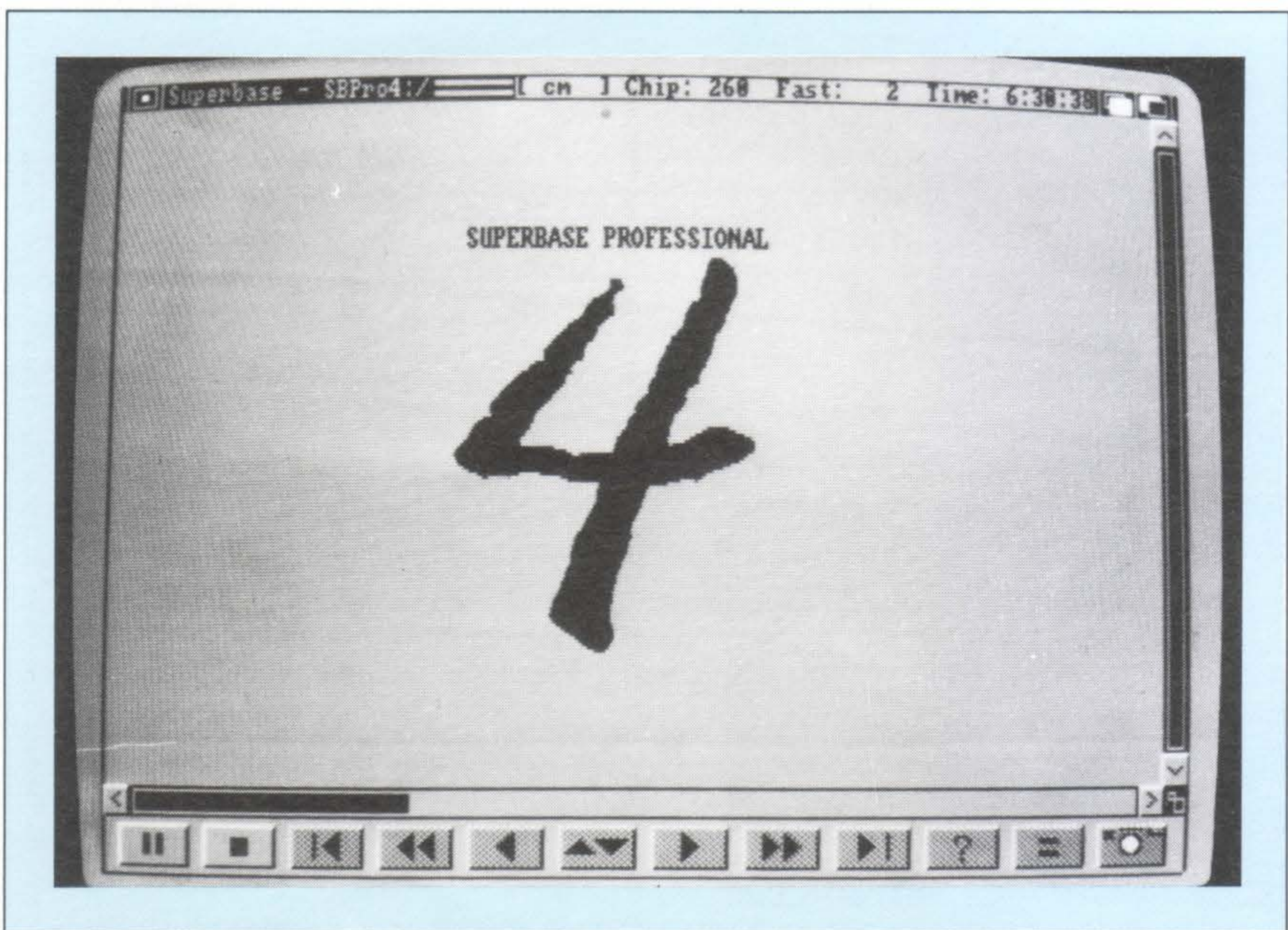
L'ambiente di lavoro di «SuperBase» è costituito da due programmi distinti: il vero e proprio **Database Manager**, ovvero il software di gestione degli archivi, ed il **Forms Designer**, tramite il quale si possono generare l'aspetto e le funzioni dell'interfaccia utente tramite la quale consultare i dati.

«SuperBase» può essere utilizzato in due modi: tramite un'interfaccia utente semplificata, basata sui classici menu a tendina di Amiga e su di una serie di gadget intuitivi nella parte bassa dello schermo che ricordano i comandi di un videoregistratore; oppure in modo diretto, nel quale l'utente digita i comandi necessari alla manipolazione dei dati.

Il programma ha infatti uno dei suoi punti di forza nel linguaggio **DML** (abbreviazione di «**Database Management Language**»): si tratta di un linguaggio molto simile nella struttura ad un tradizionale Basic opportunamente integrato con istruzioni specifiche per la gestione di archivi e di dati. La differenza fra le due modalità operative non dovrebbe essere difficile da comprendere: nel primo caso impartiamo un comando tramite il mouse, ed esso viene eseguito immediatamente; nel secondo caso una sequenza di istruzioni viene memorizzata sotto forma di programma, che potrà essere eseguito, modificato, o registrato su disco.

Se le stesse operazioni possono essere eseguite sia in modo diretto, usando il mouse, che tramite programmi, perché prendersi la briga di imparare a programmare in DML? La risposta è abbastanza semplice: perché combinando nei modi più disparati le istruzioni che il DML ci mette a disposizione è possibile ottenere dal programma prestazioni estremamente sofisticate ed automatismi difficili, se non impossibili, da ottenere in modo diretto. Spesso un programma di poche righe si rivela utile per eseguire quelle operazioni ripetitive che in modo diretto richiedono una noiosa serie di passaggi.

La disponibilità di un linguaggio interno di programmazione è infatti caratteristica comune a tutti i database relazionali di livello professionale; «SuperBase» la sfrutta al massimo, rendendo accessibili le potenzialità del DML anche attraverso il Forms Designer.

Come abbiamo detto, quest'ultimo viene utilizzato per organizzare l'aspetto dello schermo e quello della struttura di interrogazione dei dati. Sebbene un archivio possa essere consultato direttamente tramite i menu di «SuperBase», è certo comodo poter donare ai propri database un aspetto meno spartano e più intuitivo, disegnando schermi dotati di gadget e riquadri per strutturare in maniera visivamente gradevole e più semplice possibile

l'accesso ai dati. Il Forms Designer permette di caricare come sfondo un'immagine grafica in qualsiasi formato e risoluzione (compresi il modo Ham, il Dynamic Hires ed i formati GIF e PCX), e di modificarla grazie ad una serie di tool per il disegno simili a quelli di «Deluxe Paint», sebbene ovviamente meno potenti. Diventa facilissimo quindi costruire delle «maschere» per l'inserimento dei dati, sulle quali l'utente può clickare per attivare o disattivare campi logici, digitare direttamente i valori richiesti, etc. Oltre che i tradizionali menu, è possibile anche definire zone dello schermo che, una volta clickate, attivino una serie di comandi DML, rendendo così semplice la gestione e la creazione di pulsanti e gadget.

Dulcis in fundo, anche «SuperBase», come ormai la stragrande maggioranza dei programmi Amiga di recente produzione, è dotato di una porta **ARexx**, che lo rende in grado di interagire con altre applicazioni.

In sostanza, la caratteristica principale di «SuperBase Professional 4» è la flessibilità: rispetto alla versione precedente, è molto più configurabile secondo le proprie esigenze. Le restanti prestazioni più strettamente tecniche (velocità di ricerca dei dati, capacità di memorizzazione, etc.) sono rimaste sostanzialmente le stesse della release precedente, già comunque molto efficienti in questi settori. Un solo handicap, di non trascurabile entità, limita la diffusione di questo programma: il prezzo, allineato alla media di questa categoria di programmi, ovvero molto elevato (circa 450 dollari negli Stati Uniti).

Le possibilità di «SuperBase» sono però enormi, anche se lo spazio a disposizione ci ha consentito soltanto, in questa e nelle recensioni passate, di mostrare al lettore la punta dell'iceberg, e rappresenta, nel settore dei database su Amiga, il punto di riferimento in base al quale gli altri programmi devono essere valutati.

OXXI
1339 East 28th St.
Long Beach CA 90806
U.S.A.

PROTEXT 5.5

Qualche mese fa, l'annuncio che la versione Amiga del programma «**WordPerfect**» non sarebbe più stata aggiornata e supportata dalla casa produttrice ha posto fine alla possibilità di disporre, per il nostro computer, di uno dei word processor più diffusi nel mondo.

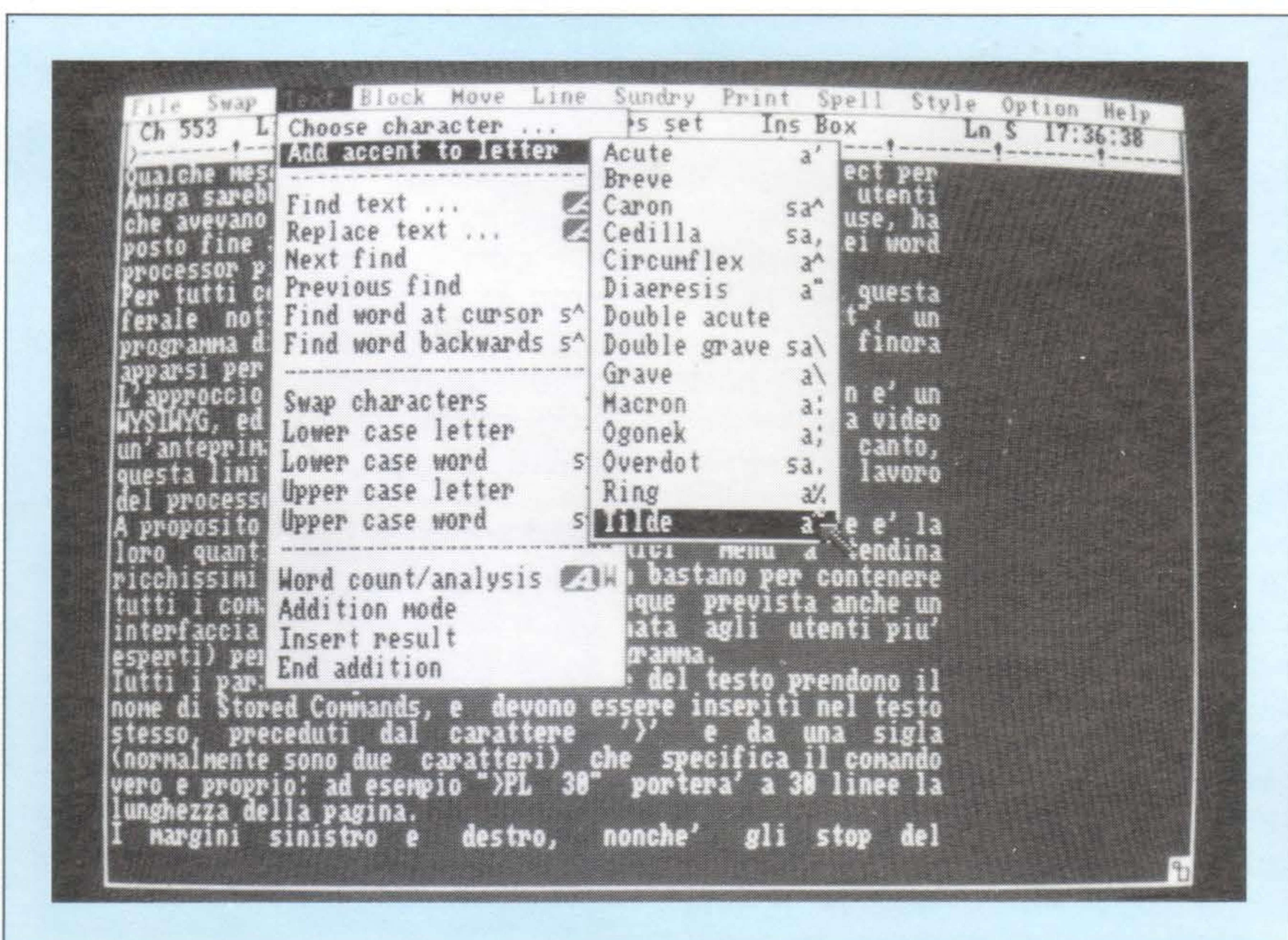
Per tutti coloro il cui cuore è stato spezzato da questa ferale notizia, l'inglese Arnor ha pubblicato «**ProText**», un software che, per origini ed impostazione, ricorda abbastanza da vicino il programma della WordPerfect Corp.: è infatti dedicato esclusivamente al trattamento di testi (senza la possibilità di gestire grafica di alcun tipo) e sfrutta in maniera abbastanza marginale l'interfaccia utente ad icone e finestre di Amiga. Sfortunatamente «ProText» divide con «WordPerfect» anche una caratteristica sgradevole: il costo molto elevato (viene venduto a 179 sterline nella sua patria d'origine).

L'approccio è di tipo tradizionale: non essendo «ProText» un word processor WYSIWYG è quindi piuttosto difficile avere a video un'anteprima di ciò che verrà stampato; d'altro canto, questa limitazione alleggerisce di molto il carico di lavoro del processore, libero di espletare altre funzioni.

A proposito di funzioni, la prima cosa che colpisce è la loro quantità: ci sono ben dodici menu a tendina ricchissimi di opzioni, ed ancora non bastano per contenere tutti i comandi disponibili. È dunque prevista anche un'interfaccia di tipo testuale (destinata agli utenti più esperti) per impartire comandi al programma.

Tutti i parametri per la formattazione del testo prendono il nome di **Stored Commands**, e devono essere inseriti nel testo stesso, preceduti dal carattere «>» e da una sigla (normalmente di due caratteri) che specifica il comando vero e proprio: ad esempio «>PL 30» imposterà a 30 linee la lunghezza della pagina.

I margini sinistro e destro, nonché gli stop del tabulatore, sono indicati dal «righello», una sequenza di caratteri (sempre presenti nella prima linea dello schermo) in cui «L» indica il margine sinistro, «!» uno stop, e «R» il margine destro. Desiderate cambiare qualcuno di questi parametri? Niente di più semplice: inserite semplicemente il nuovo righello nel testo, ricordandovi di sostituire il primo carattere con «>». Se avete già scritto del testo sotto il punto in cui avete inserito il nuovo righello, la formattazione, a video, non sarà automatica: forzate-

la con il comando **Format** del menu **Sundry**.

In alternativa, potete premere ESC (si apre la finestra del **Command Mode**) e digitare direttamente «format» seguito da Return. Questo metodo può sembrare più laborioso, ma per alcune persone, quando lavorano sulla tastiera, l'uso del mouse costituisce una distrazione o, nel migliore dei casi, una scomodità che è preferibile evitare.

Le possibilità del programma sono molto vaste: si possono avere più documenti aperti contemporaneamente, collegare logicamente più file su disco in modo che appaiano al programma come un unico documento, inserire caratteri semigrafici o lettere accentate non disponibili sulla vostra tastiera con un semplice click del mouse, aggiungere accenti, cediglie, dieresi ed altri simboli (ben quattordici in totale!) ad una lettera già inserita; oltre che le solite statistiche sul numero di parole e di lettere, potete visualizzare tutte le parole che avete utilizzato, in ordine alfabetico o in ordine di frequenza; il tutto, sull'intero documento o su una sua parte. In questo modo è facile evitare di utilizzare troppo spesso un particolare termine o modo di dire.

Il testo può essere «fissato», cioè reso insensibile a successive formattazioni (con **fix**, dato come comando o selezionato nel menu **Sundry**, per l'intero testo; con il comando **fixb** o con la voce **Fix Block** per un blocco selezionato).

«ProText» gestisce tranquillamente intestazioni, piè di pagina e note in calce: in particolare, per queste ultime occorre fissare un punto del testo al quale si riferiscono (in fase di stampa verrà aggiunto il tradizionale numerino) e definire la nota corrispondente, con «>**FN**» seguito dal testo della nota.

Quest'ultimo Stored Command può apparire ovunque nel testo, ma è consigliabile porlo sulla linea immediatamente successiva a quella in cui la nota è richiamata. Le note verranno normalmente stampate nella parte bassa della pagina (alzando il margine inferiore se necessario), ma si può anche scegliere di elencarle tutte al termine del testo.

Poiché le note vengono identificate unicamente in base alla loro posizione nella sequenza, se se ne cancella o se ne aggiunge una è fondamentale cancellare o aggiungere anche il richiamo corrispondente, altrimenti il risultato sarà terrificante: tutte le note successivamente saranno sfasate di uno.

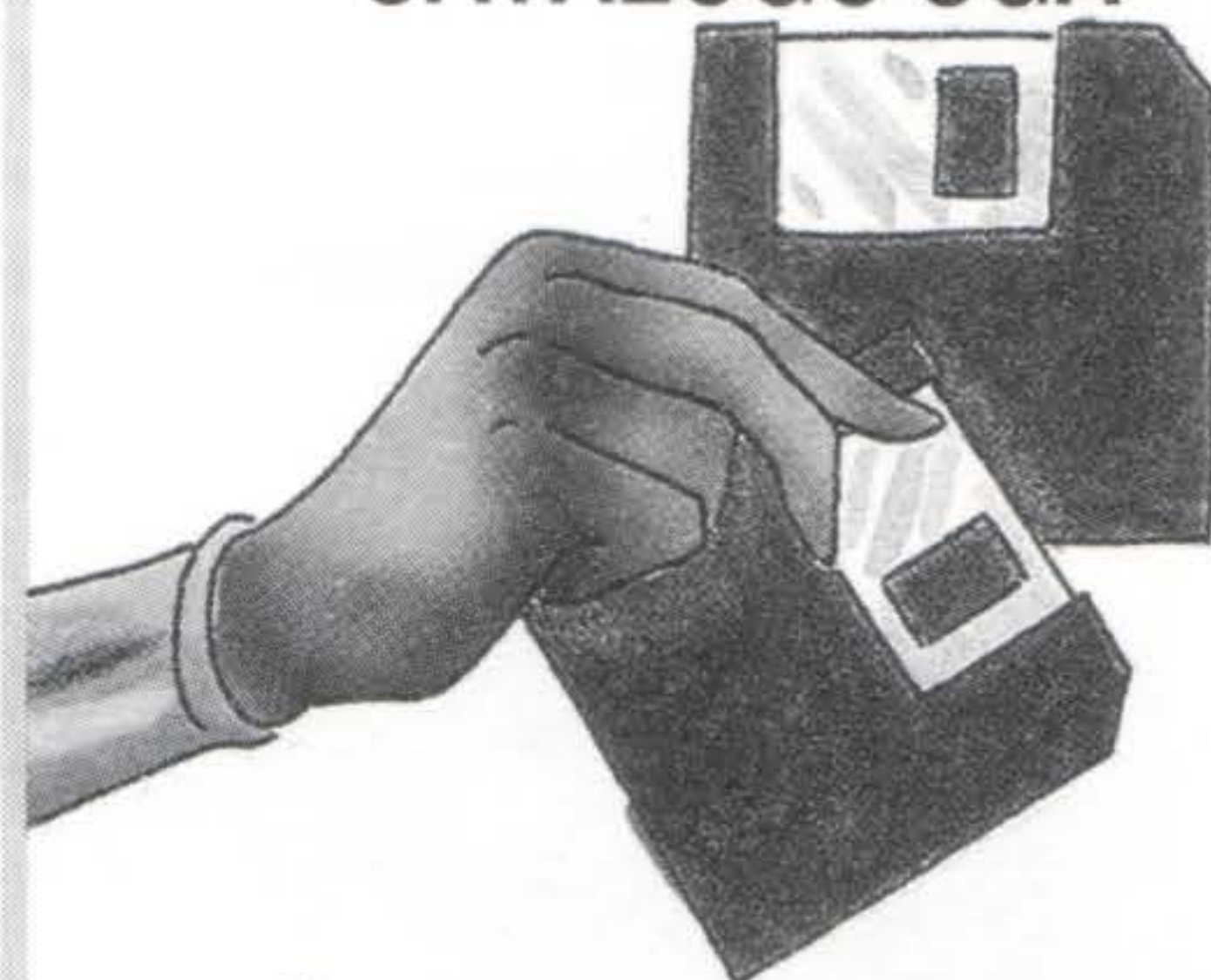
«ProText» non effettua automaticamente la divisione in sillabe: potete però inserire nelle parole un carattere speciale di controllo (**Soft Hyphen**) nei punti in cui una parola potrà, in caso di necessità, essere spezzata. Al contrario esistono **Non Break Hyphen** e **Non Break Space**, altri due caratteri speciali identici al trattino ed allo spazio tradizionali, ma che non permettono che le parole tra le quali si trovano vengano divise su righe differenti.

Come in «**Excellence**!», la generazione di indici (normali e analitici) è automatica: basta evidenziare nel testo le parole e le frasi da inserire.

Anche sotto l'aspetto del controllo ortografico «ProText» si difende davvero bene: nella versione inglese da noi provata il dizionario è il celeberrimo Collins (lo stesso usato da «**Wordworth**»); non manca il **Thesaurus**, che fornisce tutti i possibili sinonimi di una parola.

In qualsiasi momento, premendo F2 si possono effettuare operazioni sui file: copie, cancellazioni, cambiamenti di nome, formattazioni di dischi e così via. È possibile ridefinire la funzione dei tasti per rendere «ProText» compatibile con «WordStar» (noto wp disponibile su PC MsDos); per ultimo vi segnaliamo l'help in linea, ultra-sintetico (e non potrebbe essere diversamente, data la complessità del programma) ma esauriente.

Il pacchetto comprende un'utility per l'installazione su hard disk, una per la configurazione del programma (che consente di regolare tutti gli aspetti del suo funzionamento fin nei minimi particolari, dal criterio da usare per il sort delle lettere accentate al tipo di file requester), i suddetti dizionari, ed una cinquantina di drive per stampanti, differenti da quelli del WorkBench, che coprono praticamente tutti i modelli disponibili sul mercato. Con queste caratteristiche, «ProText» si colloca di diritto fra i word processor più potenti tra quelli finora realizzati per Amiga.

ARNOR
611 Lincoln Road
Peterborough
PE1 3HA England

Microbotics VXL-30

Conviene installare una scheda acceleratrice in un Amiga 500? Ecco i vantaggi e gli svantaggi apportati al nostro computer dalla presenza di un processore 68030.

di VINCENZO MARANGONI

Da quando è in circolazione Amiga 3000, molti dei possessori dei modelli 500 e 2000 sono stati afflitti da una sorta di complesso di inferiorità nei confronti del fratello maggiore, le cui prestazioni in termini di velocità rappresentano il sogno proibito di tutti coloro che fanno grafica su Amiga ma che non hanno un conto corrente bancario tale da consentire l'acquisto di un sistema 3000 completo.

Gli utenti di Amiga 2000 possono ovviare parzialmente all'inconveniente dotando la loro macchina di una tra le tante schede acceleratrici disponibili sul mercato, mentre coloro che utilizzano il più umile 500 sono ulteriormente svantaggiati dalla scarsa espandibilità della loro macchina, priva di slot nei quali inserire facilmente schede ed accessori.

Negli ultimi mesi, tuttavia, sono state molte le case produttrici di hardware che hanno proposto soluzioni in grado di accontentare coloro che, pur possedendo il modello Amiga più modesto, volessero provare il brivido della velocità.

Una tra le ultime, in ordine di tempo, è l'americana **MicroBotics** con la scheda **VXL-30**, disponibile in diverse versioni, che si differenziano per il tipo e la velocità di clock del microprocessore montato.

La versione standard monta un processore **Motorola 68030** a 25MHz, mentre quella «deluxe» ha la stessa CPU (Central Processing Unit) clockata però a 40Mhz; è inoltre disponibile una versione più economica che monta, in luogo del 68030, la sua variante **68EC030**.

La scheda VXL-030 si installa nello zoccolo del 68000 di Amiga, vicino al bus di espansione laterale.

NON C'È LA PMMU

L'unica differenza di rilievo di questa CPU rispetto alla versione standard risiede nell'assenza della **PMMU** (Programmable Memory Management Unit); si tratta comunque di una limitazione scarsamente avvertibile, come la documentazione si premura di puntualizzare, poiché il software Amiga in genere non fa direttamente uso dell'unità di gestione della memoria incorporata nei processori a 32 bit.

Le uniche eccezioni sono rappresentate dal programma di debugging «**Enforcer**», prodotto dalla Commodore stessa, e dall'utility di pubblico dominio «**SetCpu**». Ma il sistema

operativo (anche nella versione 2.0) non fa uso della MMU, e le direttive Commodore ufficiali ai programmatori consigliamo di evitarne l'uso per ragioni di compatibilità.

INSTALLAZIONE

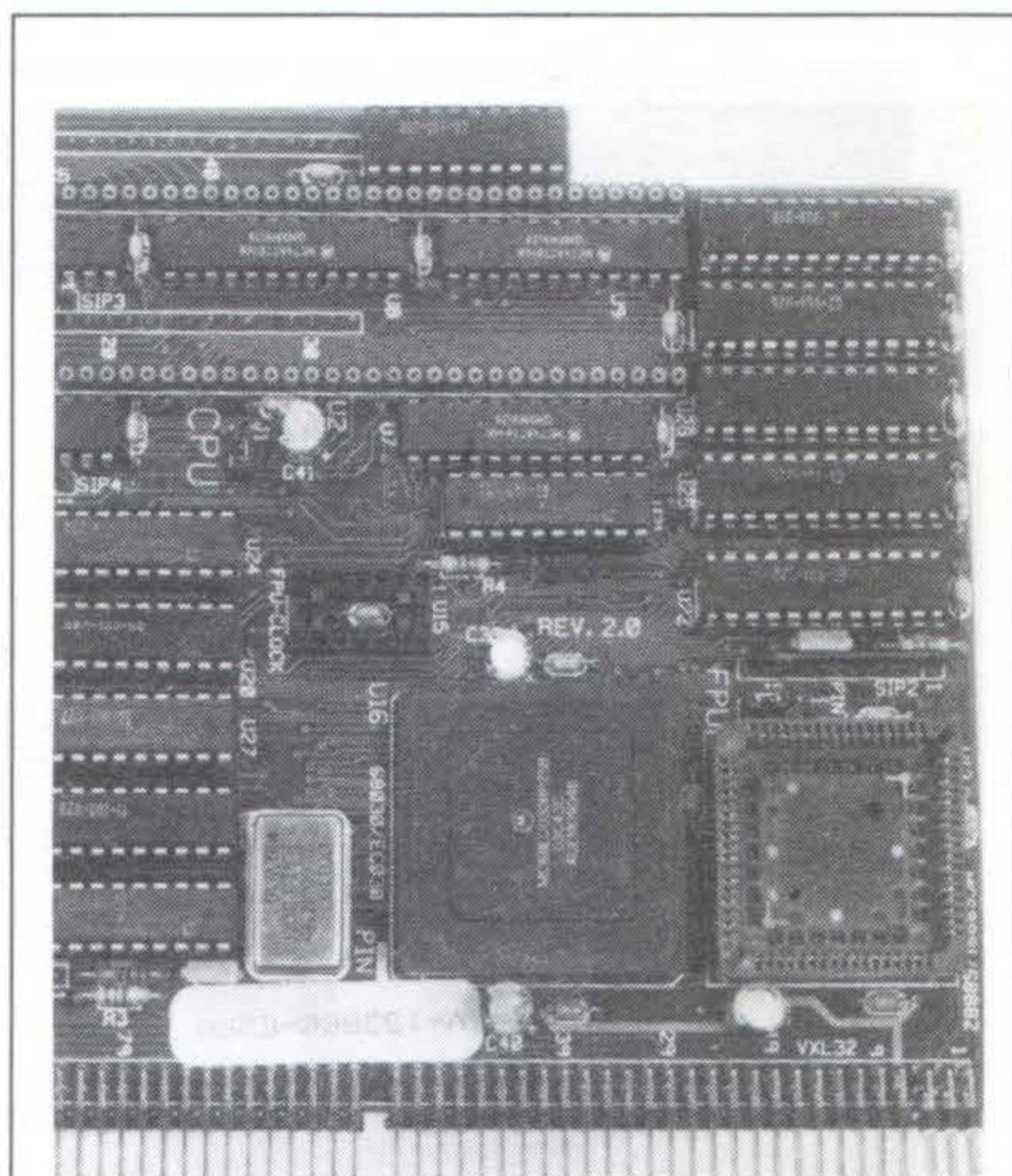
La scheda VXL-30 deve essere montata internamente ad Amiga 500 e quindi, richiedendo l'apertura del computer, rende automaticamente non valida la garanzia Commodore se l'installazione non viene effettuata da un tecnico autorizzato.

Chi comunque ha un minimo di pratica e non si lascia intimorire dall'idea di mettere le mani nelle viscere di Amiga, può tranquillamente procedere da sé all'operazione: la confezione della scheda comprende anche la speciale chiave a brugola necessaria per togliere le viti che chiudono i modelli più recenti di Amiga (per quelli più vecchi basta un cacciavite a stella).

Il processore della VXL-30 è il Motorola 68030; per assicurare la compatibilità con il software pre-esistente (che potrebbe non funzionare correttamente con una CPU a 32 bit), la procedura di installazione richiede di rimuovere il 68000 dalla piastra madre di Amiga, inserirlo nella scheda VXL-30 e montare quest'ultima nello zoccolo lasciato vuoto dal 68000. In questo modo entrambi i microprocessori coesistono sulla stessa scheda ed è possibile, via software, selezionare quello più adatto alle circostanze.

Il montaggio della VXL-30 è un'operazione delicata: occorre fare molta attenzione a non piegare i piedini del 68000 quando lo si toglie dalla motherboard e quando lo si inserisce nella VXL-30.

Le difficoltà derivano dalla scarsità di spazio disponibile all'interno di un Amiga 500, poiché è fondamentale che la scheda non tocchi in alcun punto la superficie della motherboard di Amiga.

Le istruzioni fornite dalla MicroBotics assicurano la totale compatibilità della VXL-30 con il KickStart 2.0, arrivando addirittura ad affermare che, nei modelli dotati di 68030 a 40MHz, l'utilizzo del vecchio 1.3 può portare occasionalmente a problemi di blocchi del sistema. Non siamo tuttavia riusciti a provare la VXL-30 su di un Amiga 500 Plus, in quanto la diversa disposizione dei componenti sulla motherboard del nuovo modello ne rende quasi impossibile l'inserimento.

Lo zoccolo vuoto nell'angolo inferiore destro può alloggiare il co-processore matematico 68881/68882 opzionale.

Il problema non si pone invece con la versione per Amiga 2000, nel quale l'installazione è di gran lunga più agevole.

LA COMPATIBILITÀ

Uno dei punti di forza della scheda risiede certamente nella grande compatibilità con il software preesistente, assicurata dall'utilizzo di due processori separati. La selezione della CPU da utilizzare e delle modalità d'uso si effettua via software tramite l'utility **«SetVXL»**.

Un jumper sulla scheda permette di stabilire la modalità da adottare al momento dell'accensione del computer (come default viene usato il 68030), ma digitando il comando

SetVxl 68000 Nointer

si può forzare un reboot in modo 68000 per poter caricare quei programmi (nella maggior parte dei casi giochi) che avessero problemi con la Cpu a 32 bit.

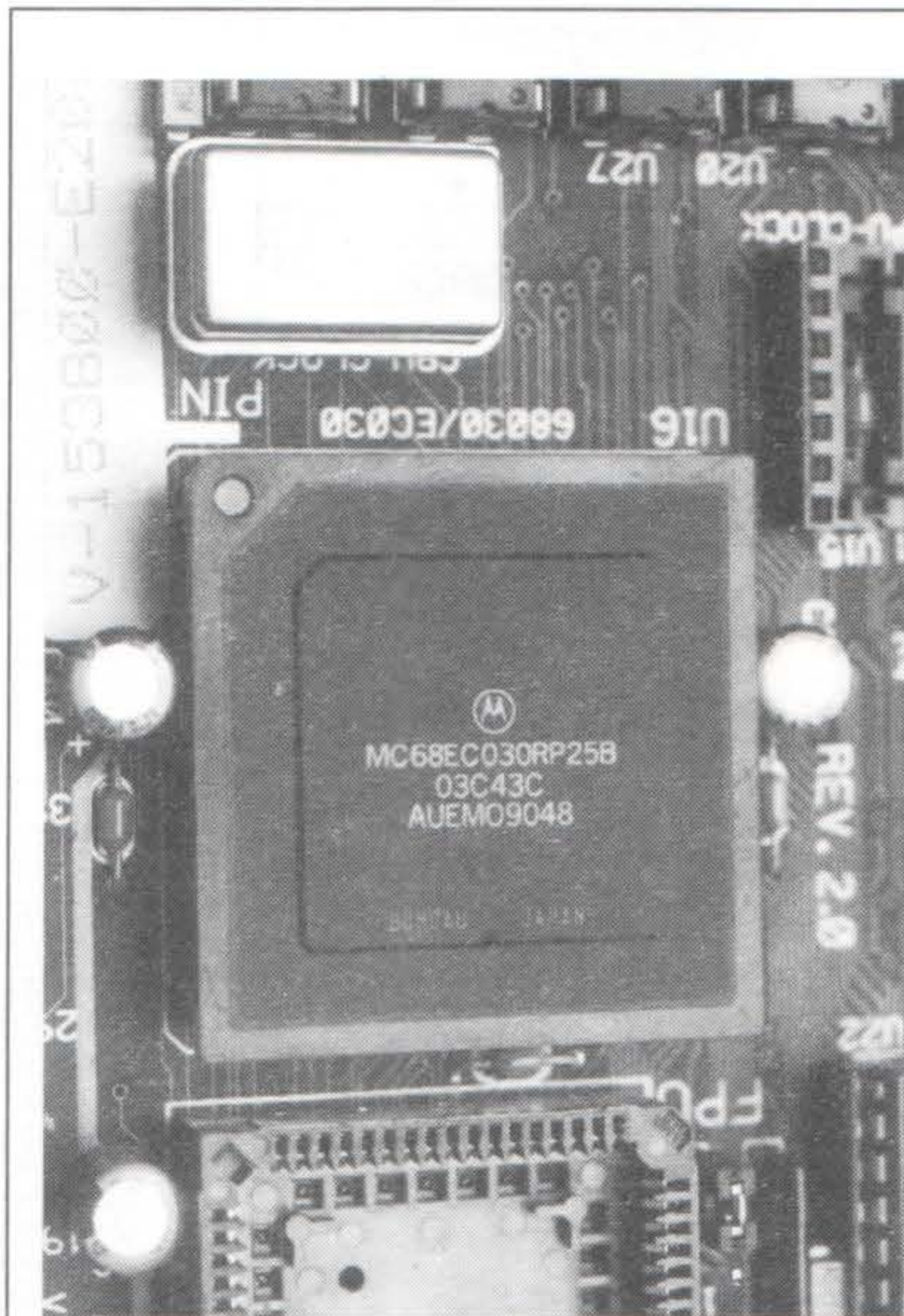
Abbiamo provato dozzine di programmi commerciali e di pubblico dominio e nessuno ha avuto problemi a funzionare con la scheda installata: solo alcuni giochi (ad esempio «RoboCod») hanno richiesto obbligatoriamente di effettuare il boot in modo 68000 per girare correttamente, mentre altri (come «Alien Breed») addirittura hanno tratto vantaggio dall'uso del 68030, funzionando meglio e più velocemente.

Sempre tramite il programma «SetVXL» si possono attivare alcune funzioni operative avanzate del 68030: il modo **Cache** e quello **Burst**.

La *cache* interna del 68030 è, in parole povere, una zona di memoria riservata all'interno del processore nella quale vengono temporaneamente mantenuti le istruzioni eseguite ed i dati trattati. In questo modo, dovendo ripetere sequenze di istruzioni uguali, la CPU non ha bisogno di dover accedere alla Ram esterna (notoriamente più lenta) e può quindi effettuare i calcoli necessari più rapidamente.

Il modo Burst è una caratteristica operativa propria dl 68030, nella quale il processore può accedere alla memoria leggendo quattro longword consecutive alla volta invece di una. Il vantaggio del Burst Mode è un incremento di velocità, ottenibile però soltanto se la scheda viene utilizzata in unione a Ram a 32 bit.

LE PRESTAZIONI

La memoria normalmente presente all'interno di Amiga è infatti a 16 bit, essendo il 68000 una CPU a 16 bit; il 68030 è ovviamente in grado di utilizzare la stessa ram, ma le prestazioni in termini di velocità ri-

Il 68EC030, una versione più economica del 68030 standard, opera a 25 MHz ma è privo della PMMU (Programmable Memory Management Unit).

Secondo «SysInfo», la velocità di un Amiga equipaggiato con la VXL-030 senza ram a 32 bit equivale ad una volta e mezzo quella di un 500 standard.

sultano penalizzate rispetto all'utilizzo di memoria a 32 bit.

Per questo motivo l'ideale sarebbe impiegare la VXL-30 in unione all'apposita espansione di memoria a 32bit della MicroBotics (VXL-RAM 32): in questo modo, oltre che poter usufruire dei vantaggi del modo Burst, si può copiare il contenuto della Rom di Amiga nella ram veloce a 32 bit (con l'opzione FastRom di «SetVXL»), migliorando ulteriormente la velocità del sistema.

Anche con la ram a 16 bit di Amiga la presenza della scheda si fa sentire, sebbene in misura meno evidente: mediamente l'aumento di velocità è di una volta e mezza rispetto al normale, con applicazioni che non fanno uso del set di istruzioni specifico del 68030 (la maggioranza dei programmi). Nei casi in cui i programmi siano disponibili in versione per processori a 32 bit (come capita con molti pacchetti di grafica e di ray tracing) l'incremento è ovviamente superiore, specialmente se nella scheda viene montato il coprocessore matematico opzionale 68881, arrivando fino a ridurre di un quinto i tempi di elaboazione.

Dotando la scheda di ram a 32 bit, il 68030 non è più costretto a sfruttare il controller della memoria interno di Amiga (tarato per un processore a 16 bit ed a 7,16 MHz), che rappresenta così il «collo di bottiglia» che frena le prestazioni del sistema. Ne deriva che l'acquisto della sola VXL-30 non ha molto senso, se non prevedendo di dotarla dell'apposita espansione e, possibilmente, della FPU (Floating Point Unit, il coprocessore matematico).

NE VALE LA PENA?

La domanda che viene spontanea esaminando le caratteristiche della VXL-30 (ma gli stessi dubbi sono legittimi riguardo a qualsiasi scheda acceleratrice per Amiga 500) è: vale la pena di comprarla?

La risposta, considerato il rapporto prezzo/prestazioni della scheda nuda e cruda è: francamente no. L'incremento di velocità apportato dalla VXL-30 ad un Amiga 500 standard non è certo stupefacente, e non risulta proporzionale alla spesa richiesta per l'acquisto (630.000 lire per la versione a 25MHz e 960.000 lire per quella a 40MHz).

Le cose cambiano dotando la scheda di almeno due mega di memoria a 32 bit: in questo caso l'incremento di velocità arriva fino a sette volte (ed in alcuni casi anche di più, se si aggiunge il coprocessore matematico) e la differenza si sente eccome.

In definitiva, ci sentiamo di consigliare l'acquisto della scheda soltanto a chi ha deciso di preventivare contemporaneamente (o in un futuro molto prossimo) anche la spesa per un'espressione di memoria a 32 bit: la somma totale resta sempre inferiore a quella richiesta da un Amiga 3000, ed il miglioramento delle prestazioni del computer è indubbiamente sensibile. □

Giochiamo a dadi con ARexx

Prendiamo confidenza con il linguaggio implementato come standard nel nuovo sistema operativo 2.0, che rivela una potenza insospettata

di LUCA MIRABELLI
Prima parte

*Da molti anni a questa parte, non c'è stato micro, personal o home computer che non includesse, nella dotazione base di software, un **interprete del linguaggio Basic**.*

I motivi del successo di questo linguaggio sono davanti agli occhi di tutti: facilità di apprendimento e discreta potenza. Non è azzardato stimare che la stragrande maggioranza dei programmatori conosca ed abbia avuto occasione di utilizzare questo linguaggio (salvo poi dirne in giro tutto il male possibile se così impone la moda, come è avvenuto negli ultimi anni).

*Lungi da noi l'idea di criticare il BASIC in sé, dunque; ma per quanto riguarda **AmigaBasic** il discorso è assai differente. Questo interprete, scritto da MicroSoft e presente sul disco Extras fino alla penultima versione del WorkBench (1.3), è infatti un prodotto di qualità mediocre, che sfrutta solo una minima parte delle caratteristiche della macchina sulla quale gira: inoltre, la velocità di una lumaca è supersonica, se confrontata alla sua.*

Le puntuali voci circa una nuova versione di AmigaBasic (l'ultima release risale ormai al 1986) non hanno mai trovato conferma: è dunque giustificata la scelta, da parte di Commodore, di non fornire più l'interprete con le macchine in vendita.

*In sostituzione è stato scelto (a partire dalla release 2.0 del sistema operativo) il **linguaggio ARexx**, sconosciuto ai più, ma molto più adatto del BASIC ad un computer come Amiga, linguaggio del quale «AmigaByte» si è a suo tempo occupata sul fascicolo 23.*

Nato con il nome di «Rexx» per girare su grossi mainframe di marca IBM, è stato portato su Amiga (con ottimi risultati) da William Hayes, ed in seguito acquisito dalla Commodore-Amiga Inc.

ARexx presenta alcuni aspetti che ricordano molto da vicino il BASIC, ai quali però unisce il vantaggio principale del C (ovvero l'uso di librerie di funzioni) insieme ad una caratteristica decisamente unica: il concetto di porta e di comunicazione inter-processo, che consente di sfruttare al meglio le avanzate capacità di multitasking tipiche di Amiga.

PRONTI? VIA!

*Tutto ciò di cui avete bisogno è un interprete ARexx (gli utenti di WorkBench 1.3 o precedenti dovranno procurarselo separatamente), ed un editor per la modifica dei testi (vanno bene anche «**Ed**» o «**MicroEmacs**», presenti rispettivamente sul disco WorkBench e sul disco Extras).*

*Per prima cosa attivate l'interprete, usando il programma «**RexxMast**»: questo compito viene solitamente*

```
/* Simula un lancio di dadi */
say time(s)

cancella = d2c(27) || "[0;0H" || d2c(27) || "[J"
say cancella

/* richiede il numero di giocatori */

say "Introdurre il numero di giocatori:"
parse pull g

/* memorizza i nomi dei giocatori */

do x=1 to g
        say "Giocatore numero" x || ", inserisci il tuo nome:"
        parse pull nome.x
end

/* simula il lancio dei dadi */

say cancella
do x=1 to g
        say nome.x || ", premi <RETURN> per tirare i dadi!"
        parse pull
        d1=random(1,6); d2=random(1,6)
        say d1 "+" d2 "=" d1+d2
        punteggio.x=d1+d2
end

/* confronta i punteggi e stabilisce il vincitore */

massimo=0
do x=1 to g
        if punteggio.x>massimo then do
                massimo=punteggio.x
                vincitore=nome.x
        end
end
say; say "Il vincitore e`" vincitore || ", con un punteggio di" massimo
```

Fig. 1

assolto in modo del tutto automatico nella Startup-Sequence del WB 2.0, perciò non dovrete preoccuparvene.

*Portatevi, con il comando CD, nella directory «ARexx» del dischetto allegato a questo numero, e digitate «**rx dadi**» per avviare il primo programma, tenendo sott'occhio il listato riportato in figura 1.*

*Esaminiamolo passo passo: la prima linea contiene un commento, racchiuso tra le coppie di caratteri «**/***» e «***/**». Non fatevi spaventare dalle due istruzioni successive: la loro funzione è semplicemente quella di cancellare lo schermo (o meglio la finestra Shell attuale), utilizzando la sequenza speciale di caratteri **<esc>[0;0H<esc>[J** riportata sul manuale AmigaDOS.*

LE VARIABILI STRINGA

*In pratica, con la prima istruzione abbiamo definito la variabile stringa **cancella** (ricordiamo ai meno esperti che una stringa non è altro che una sequenza di caratteri qualsiasi) in modo da farle contenere la sequenza suddetta. Le variabili stringa, a differenza di quanto succede nel BASIC, non hanno alcun particolare suffisso. In ARexx, infatti, è lecito assegnare ad una variabile qualsiasi tipo di valore (intero, reale o stringa); eventuali problemi derivanti dai contenuti delle variabili saranno segnalati al momento dell'impiego, ad esempio se cercheremo di moltiplicare due versetti della Divina Commedia.*

*Poiché la maggior parte degli editor non permette di inserire il carattere <esc> nel testo, abbiamo utilizzato la funzione **d2c(n)**, che restituisce il carattere con codice **n**. Poiché il codice di <esc> è 27, possiamo aggirare elegantemente il problema con **d2c(27)**; per concatenare più stringhe abbiamo utilizzato l'operatore «|» (è quello presente sulla tastiera di Amiga sopra al carattere «\», in alto a destra).*

*L'istruzione successiva, **say**, è l'equivalente della cara, vecchia PRINT del Basic: si occupa, cioè, di visualizzare sullo schermo stringhe di testo o contenuti di variabili, come nel nostro caso: ricordiamo che la «visualizzazione» della sequenza speciale **cancella** ha la funzione di ripulire lo schermo.*

*Idem dicasi per la successiva **say**, che stampa un messaggio richiedendo il numero dei giocatori.*

```
Introdurre il numero di giocatori:
Abcde
Introdurre il numero di giocatori:
3
Giocatore numero 1, inserisci il tuo nome:
Amiga
Giocatore numero 2, inserisci il tuo nome:
Atari
Giocatore numero 3, inserisci il tuo nome:
Macintosh

Amiga, premi <RETURN> per tirare i dadi!
5 + 4 = 9
Atari, premi <RETURN> per tirare i dadi!
3 + 4 = 7
Macintosh, premi <RETURN> per tirare i dadi!
3 + 5 = 8

Il vincitore e` Amiga, con un punteggio di 9
```

DATI IN INGRESSO

*L'input, ovvero l'inserimento dei dati da parte dell'utente, è una parte molto importante di tutti i programmi di questo mondo; in ARexx il compito è svolto dall'istruzione **parse**, che qui vediamo nella forma **parse pull g**, che immagazzina ciò che viene digitato da tastiera (in questo caso, il numero dei giocatori) nella variabile **g**.*

*La presenza della parola chiave **do** ci rivela che quello che segue è un gruppo di istruzioni, impressione confermata dall'altra parola chiave **end** che si trova alla fine. L'effettivo utilizzo delle istruzioni del gruppo dipende dal contesto nel quale si trova il **do**: nel nostro caso, **do x=1 to g** significa «poni la variabile **x** a 1, e ripeti il ciclo, incrementandola ogni volta, finché non avrà raggiunto il valore **g**". In pratica, le istruzioni vengono ripetute **g** volte, una per ogni giocatore.*

*Vediamole, queste istruzioni: la prima **say** stampa il messaggio «Giocatore xxx, inserisci il tuo nome:», ed è un esempio di come testo e variabili possano essere combinati all'interno di una stessa **say**. C'è di più: nel caso in cui si desideri concatenare due stringhe interponendo uno spazio, si può fare a meno dell'operatore «||», come abbiamo fatto tra «Giocatore» ed x.*

*La successiva **parse pull** è identica alla precedente, ma la variabile che ha per oggetto merita un'attenzione particolare: si tratta infatti di una variabile composta, come indica il punto nel suo nome.*

*Quando ARexx incontra una variabile di questo tipo, la seconda parte del nome viene valutata: nel nostro caso, il nome di variabile x verrà sostituito con il valore attuale. In questo modo ci è possibile definire una famiglia di variabili **nome.1**, **nome.2**, **nome.3...** e richiamare ciascuna di esse con il medesimo nome, ma con un diverso indice numerico: chi conosce il BASIC, sappia che stiamo parlando dell'equivalente degli array. Un po' di terminologia non guasta, soprattutto per comprendere meglio i messaggi di errore: la prima parte del nome di queste variabili, precedente il numero, è detta **stem**.*

```
/* Simula un lancio di dadi - Versione migliorata e corretta */

cancella = d2c(27) || "[0;0H" || d2c(27) || "[J"
say cancella

/* richiede il numero di giocatori (e che sia un numero!) */

do while datatype(g,"N")=0
        say "Introdurre il numero di giocatori:"
        parse pull g
end

/* memorizza i nomi dei giocatori */

do x=1 to g
        say "Giocatore numero" x || ", inserisci il tuo nome:"
        parse pull nome.x
end

/* simula il lancio dei dadi */

d1=random(1,6,time("E")*100); d2=random(1,6,time("E")*100)

say cancella
do x=1 to g
        say nome.x || ", premi <RETURN> per tirare i dadi!"
        parse pull
        d1=random(1,6,time("E")*100); d2=random(1,6,time("E")*100)
        say d1 "+" d2 "=" d1+d2
        punteggio.x=d1+d2
end

/* confronta i punteggi e stabilisce il vincitore */

massimo=0
do x=1 to g
        if punteggio.x = massimo then vincitore = vincitore compress(nome.x)
        if punteggio.x > massimo then do
                massimo=punteggio.x
                vincitore=compress(nome.x)
        end
end
say;

/* se il vincitore e` unico... */

if words(vincitore)=1 then do
        say "Il vincitore e`" vincitore || ", con un punteggio di" massimo
        exit
end

/* se invece i vincitori sono piu` di uno... */

say "I vincitori sono:"
say vincitore
say "con un punteggio di" massimo
```

Fig. 2

ALEA IACTA EST

*Il ciclo è ora terminato: ripuliamo lo schermo dalla spazzatura presente (basterà un altro **say cancella**),e diamo inizio al gioco vero e proprio.*

*La prima **say** invita il giocatore di turno, chiamandolo per nome, a premere Return per lanciare i dadi; la successiva **parse pull** è senza argomenti, e non si tratta di un errore di stampa. Il motivo è semplice: non ci interessa affatto che quanto eventualmente digitato dal giocatore venga memorizzato in qualche variabile, e l'istruzione è soltanto un modo come un altro per attendere la pressione del Return.*

*La linea successiva attribuisce alle variabili **d1** e **d2** due valori casuali tra uno e sei, facendo uso della funzione **random(min,max)**, che restituisce un numero a caso compreso nell'intervallo specificato: notiamo inoltre come si possano porre più istruzioni su di una stessa linea, separandole con un punto e virgola.*

*Dopo avere visualizzato (ancora con **say**) il risultato del lancio dei dadi, il punteggio ottenuto viene memorizzato in una nuova famiglia di variabili, la **punteggio.x**.*

*Terminato il ciclo, la variabile **massimo** viene posta a zero: durante il ciclo successivo, il punteggio di ciascun giocatore viene confrontato con il massimo: **if punteggio.x > massimo then do** stabilisce che il blocco di istruzioni (fino all'**end**) debba essere eseguito soltanto se la condizione compresa tra **if** e **then** è verificata. In tal caso il nome del giocatore viene copiato nella variabile **vincitore**, e la variabile **massimo** viene aggiornata al nuovo punteggio da battere, il suo.*

*Al termine del ciclo, tutti i punteggi saranno stati confrontati, e **vincitore** e **massimo** conterranno informazioni attendibili. Occupiamoci ora dell'annuncio al mondo dell'esito della competizione.*

*Una **say** senza argomenti serve soltanto per lasciare una riga vuota, mentre l'ultima istruzione del programma visualizza i dati che abbiamo tanto faticosamente elaborato.*

I DADI SONO TRUCCATI...

Inestimabile dal punto di vista didattico, il nostro programma è però un fallimento quasi totale sotto l'aspetto del divertimento: è sufficiente rispondere alla domanda sul numero dei giocatori con una lettera per generare un messaggio di errore e terminare prematuramente l'esecuzione; eventuali ex-aequo non sono inoltre considerati (viene scelto come vincitore il primo concorrente ad avere ottenuto quel punteggio). Dulcis in fundo, e questa è una cosa che nessun serio giocatore d'azzardo potrebbe mai tollerare, i dadi sono truccati: vi sarete accorti che i numeri usciti sono gli stessi in tutte le partite!

*Vediamo di aggiustare qua e là il **listato «Dadi»** per porre rimedio ai problemi evidenziati, approfittandone (ne dubitavate?) per fare la conoscenza di qualche altro aspetto di ARexx: vi anticipiamo che potete trovare tutte le correzioni nel **listato «Dadi2»**.*

*Come fare per controllare che il valore immesso sia effettivamente un numero? Ci viene in aiuto la funzione **datatype(stringa,"N")**, dove «N» sta per «Numeric», che ritorna 1 o 0 a seconda che la stringa sia o meno un numero. Piazzando opportunamente un **do while datatype (g,"N")=0** ed un **end**, possiamo far ripetere la richiesta relativa al numero di giocatori finché l'input non sarà soddisfacente: è proprio questa la funzione di **do while**.*

TABS	ARG	B2C	C2B
C2D	C2X	D2C	D2X
EOF	MAX	MIN	POS
X2C	X2D	DATE	FIND
FOR	FUZZ	HASH	LEFT
OPEN	SEEK	SHOW	SIGN
TIME	TRIM	WORD	BITOR
CLOSE	INDEX	LINES	RANDU
RIGHT	SPACE	STRIP	TRACE
TRUNC	UPPER	VALUE	WORDS
ABBREV	ADDLIB	BITAND	BITCHG
BITCLR	BITSET	BITTST	BITXOR
CENTER	CENTRE	COPIES	DELSTR
DIGIT	EXISTS	EXPORT	IMPORT
INSERT	LENGTH	PRAGMA	RANDOM
READCH	READLN	REMLIB	SUBSTR
SYMBOL	VERIFY	XRANGE	ADDRESS

Elenco delle funzioni di base dell'interprete, estratto dalla rexxsyslib.library.

BITCOMP	COMPARE	DELWORD
GETCLIP	LASTPOS	OVERLAY
REVERSE	SETCLIP	STORAGE
SUBWORD	WRITECH	WRITELN
COMPRESS	DATATYPE	GETSPACE
ERRORTEXT	FREESPACE	TRANSLATE
WORDINDEX	SOURCELINE	WORDLENGTH

LE FUNZIONI DESCRITTE

words()

Sintassi: words(stringa)

restituisce il numero di parole di cui stringa è composta.

random()

Sintassi: random(min,max,seme)

restituisce un numero pseudocasuale tra min e max, utilizzando seme, se presente, come base per la generazione.
La differenza tra max e min non può essere superiore a 1000.
Nota: i risultati NON sono distribuiti uniformemente!

compress()

Sintassi 1: compress(stringa)

restituisce stringa privata degli spazi: compress ("Ciao a tutti") --> "Ciaotutti"

Sintassi 2: compress(stringa,lista)

restituisce stringa privata dei caratteri compresi nella stringa lista; ad esempio compress(Promessisposi,".,:;()–") restituisce il contenuto della variabile Promessisposi privato della punteggiatura.

d2c()

Sintassi: d2c(n)

restituisce il carattere che ha codice ASCII n. La tabella dei codici ASCII è riportata sul manuale del computer.

datatype()

Sintassi 1: datatype(stringa)

restituisce la stringa "NUM" se stringa contiene un numero, "CHAR" in caso contrario.

Sintassi 2: datatype(stringa,test)

controlla se la stringa è del tipo desiderato; restituisce 1 in caso di esito positivo, altrimenti 0.

Il parametro test può essere uno dei seguenti:

"A"	. . . .	Stringa di caratteri alfanumerici
"M"	. . . .	Caratteri alfabetici
"U"	. . . .	Caratteri alfabetici maiuscoli
"L"	. . . .	Caratteri alfabetici minuscoli
"N"	. . . .	Numero
"W"	. . . .	Numero intero
"B"	. . . .	Stringa di cifre binarie (0 e 1)
"X"	. . . .	Stringa di cifre esadec. (0...F)
"S"	. . . .	Nome di variabile definita

time()

Sintassi 1: time()

restituisce una stringa contenente l'ora esatta nella forma HH:MM:SS

Sintassi 2: time(opzione)

restituisce valori differenti a seconda dell'opzione:

"C"	. . .	ora esatta alla anglosassone (AM/PM)
"E"	. . .	tempo trascorso [dall'avvio del programma o dall'ultima time ("R")]
"H"	. . .	ore trascorse dalla mezzanotte
"M"	. . .	minuti trascorsi dalla mezzanotte
"S"	. . .	secondi trascorsi dalla mezzanotte
"R"	. . .	azzera il timer utilizzato per time ("E")

SARÀ UN CASO?

Il problema dei dadi truccati richiede, forse, una spiegazione più tecnica.

*I computer ignorano cosa sia il caso, ed in situazioni come questa ricorrono alla generazione di numeri pseudo-causali. In pratica, partendo da un numero detto **seme**, per mezzo di calcoli vari si arriva ad un numero che non ha apparentemente alcuna connessione con il seme che l'ha generato. Purtroppo, il seme ha sempre lo stesso valore ogni volta che abbiamo il programma, e questo dà origine all'inconveniente che abbiamo riscontrato.*

Se solo potessimo cambiare il seme, anche di poco o con regolarità, fra tutti i lanci di dadi, le sequenze generate sarebbero completamente diverse. Il modo più comune per risolvere la questione consiste nel ricorrere, come seme, ad un valore fornito dal timer del computer.

*A questo scopo utilizziamo la funzione **time("E")**, (la «E» significa «Elapsed», trascorso) che ritorna il tempo (in secondi) passato dall'avvio dell'interprete.*

Questi sono tutti i file che dovete copiare sul dischetto WorkBench 1.3 per installare correttamente ARexx.

Poiché il seme deve essere un intero, e il risultato di time("E") ha due numeri decimali, lo moltiplichiamo per 100 ed il gioco è fatto. Per far sì che ARexx lo utilizzi come nuovo seme, poniamolo come terzo parametro nella chiamata alla funzione **random()**; inoltre, dato che anche in questo modo i risultati del primo giocatore continuano ad essere sempre gli stessi, richiamiamo la **random()** a vuoto un paio di volte prima di iniziare la competizione.

Siamo ancora ben lontani dalla simulazione di un lancio di dadi (noterete, ad esempio, che il 6 e l'1 escono molto più raramente degli altri numeri), ma per i nostri scopi la modifica è sufficiente.

Per quanto riguarda gli ex-aequo, uno dei modi per risolvere il problema potrebbe essere il seguente: per ogni giocatore, prima di vedere se il suo punteggio è maggiore del massimo, controlliamo se per caso è uguale al massimo. In tal caso non diventa il nuovo campione, ma il suo nome viene aggiunto alla variabile **vincitore**, che conterrà così il nome di tutti i giocatori che hanno ottenuto il punteggio **massimo**.

Anche la parte che si occupa della stampa dei risultati necessita di qualche modifica: occorre utilizzare due frasi differenti per l'annuncio, a seconda che i vincitori siano uno o più di uno.

Il metodo più semplice consiste nel contare il numero di parole di **vincitore**, e regolarsi di conseguenza: così facendo, però la vostra amica Maria Luisa rischia di vedere ogni sua vittoria annunciata come un pareggio...

Risolviamo la questione togliendo fin dall'inizio tutti gli spazi dai nomi dei giocatori, sostituendo nell'ultima parte del programma **nome.x** con **compress(nome.x)**. La funzione **compress(stringa)**, infatti, restituisce proprio la **stringa** privata degli spazi.

Adesso possiamo dunque utilizzare la funzione **words(vincitore)**, il cui valore è pari al numero di parole nella stringa **vincitore**: se è uguale ad uno stampiamo il messaggio di prima ed usciamo dal programma (con **exit**); altrimenti visualizziamo «**I vincitori sono:**», l'elenco dei vincitori, ed il loro punteggio.

ADESSO TOCCA A VOI!

Provate ora a modificare il programma secondo le vostre esigenze: ad esempio, cambiate le modalità di stampa dei risultati (magari introducendo qualche rozzo elemento grafico). Noi lo abbiamo fatto in «**Dadi3**» (che potete trovare sempre sul dischetto allegato a questo fascicolo).

Ricordatevi sempre di:

— lavorare su di una copia del file per non perdere quello originale;

— lasciare sempre un commento nella prima riga, altrimenti il listato non sarà riconosciuto come programma ARexx;

— non lasciare mai spazi tra il nome di una funzione e la parentesi aperta (pena errori di varia natura);

- inizializzare tutte le variabili prima dell'utilizzo: a differenza del Basic, una variabile mai usata in precedenza **NON** ha valore zero, ma viene scambiata per una stringa;

— far corrispondere ad ogni **do** un appropriato **end**, magari spostando le istruzioni del gruppo di qualche spazio verso destra per maggiore chiarezza.

Scrivete voi stessi qualche programmino, utilizzando le istruzioni descritte finora e le funzioni sviscerate dettagliatamente nel riquadro: nella prossima puntata ne incontreremo molte altre, esamineremo il comportamento dell'interprete in modo più rigoroso, e vedremo un primo esempio di interazione con altri programmi. Arrivederci!

LE VOSTRE IDEE, I VOSTRI PROGRAMMI

La redazione di Amiga Byte invita tutti quelli di voi che hanno idee o programmi già realizzati ad inviarceli in visione o a proporceli. I programmi debbono essere su dischetto, completi di istruzioni, ed assolutamente originali, ovvero pensati e realizzati da voi. Mandate sempre una copia perché tutto il materiale, pubblicato o meno, non viene restituito. Qualunque programma pubblicato verrà compensato previo accordo con l'autore. Indirizzate i vostri lavori ad Amiga Byte, c.so Vitt. Emanuele 15, 20122 Milano.

(segue da pag. 26)

BlockPerTracks è un'altra caratteristica specifica dell'hard disk: nel caso in esame i blocchi per traccia sono 33. Si può calcolare la capacità di ciascuna traccia dal momento che ogni blocco contiene 512 byte (33 x 512 = 16896 byte).

Reserved contiene il numero di blocchi riservati al sistema operativo per i suoi usi (solitamente 2). Questi blocchi sono i settori di boot usati da AmigaDos durante la formattazione, e sono piuttosto importanti dal momento che sono il luogo prediletto da alcuni virus per infettare l'hard disk.

COSA SIGNIFICA INTERLEAVE?

Il fattore di **Interleave** merita un discorso a parte: se il disco gira molto velocemente, può accadere che il controller non riesca a trasferire i dati dei vari settori alla velocità alla quale questi dati passano sotto la testina: per questo motivo, per leggere due settori consecutivi occorrerebbe impiegare due rotazioni complete del disco. Uno spreco di tempo inaccettabile.

L'interleave è un artificio con il quale i settori numerati consecutivamente non sono adiacenti, ma intervallati opportunamente da altri *n* settori, dove *n* è il valore di Interleave,

ed è tanto più alto quanto più lento è il controller.

LowCyl e **HighCyl** definiscono la grandezza della partizione. Il valore di **LowCyl** potrebbe essere 0, ma la routine di autoboot del sistema operativo 1.3 riserva questo cilindro per sé. Il valore di **HighCyl** dipende dalle caratteristiche fisiche dell'hard disk. La nostra partizione contiene 670 + 1 = 671 cilindri per un totale di 671 x 67584 = 45348864 byte, pari a poco più di 43 Mbyte. Queste informazioni sono utili per creare altre partizioni (o variare le dimensioni di quelle esistenti): tutto quello che è necessario fare è modificare i valori numerici di **Low** e High Cylinder avendo naturalmente cura di non sovrapporre due o più partizioni.

Anche **Mount** è interessante: settando questo valore ad 1, l'icona associata alla partizione apparirà automaticamente non appena AmigaDos vedrà, con l'istruzione Mount, l'hard disk; in caso contrario l'icona non apparirà finché non si accederà al disco per la prima volta.

Altra parola chiave importante è **Buffers**. Il suo valore (in questo caso 32) comunica ad AmigaDos quanta memoria riservare come *cache* all'accesso della partizione. La regola generale recita che quanto più alto risulta questo valore, tanto più veloce è l'accesso alla partizione.

Globvec identifica il tipo di device drive. In questo caso −1 sta ad indicare un programma scritto in C o in assembler.

BufMemType indica il tipo di memoria usata per il buffer (0 e 1 = qualsiasi; 2 e 3 = CHIP; 4 e 5 = FAST). **Mask** e **MaxTransfer** indicano rispettivamente gli indirizzi di memoria ai quali il controller può accedere direttamente, ed il numero massimo di blocchi trasferiti per ogni accesso. Qualora si manifestassero inconvenienti nel trasferimento di grossi file tra due HD montati nella stessa macchina, può risultare utile abbassare il valore di quest'ultimo parametro.

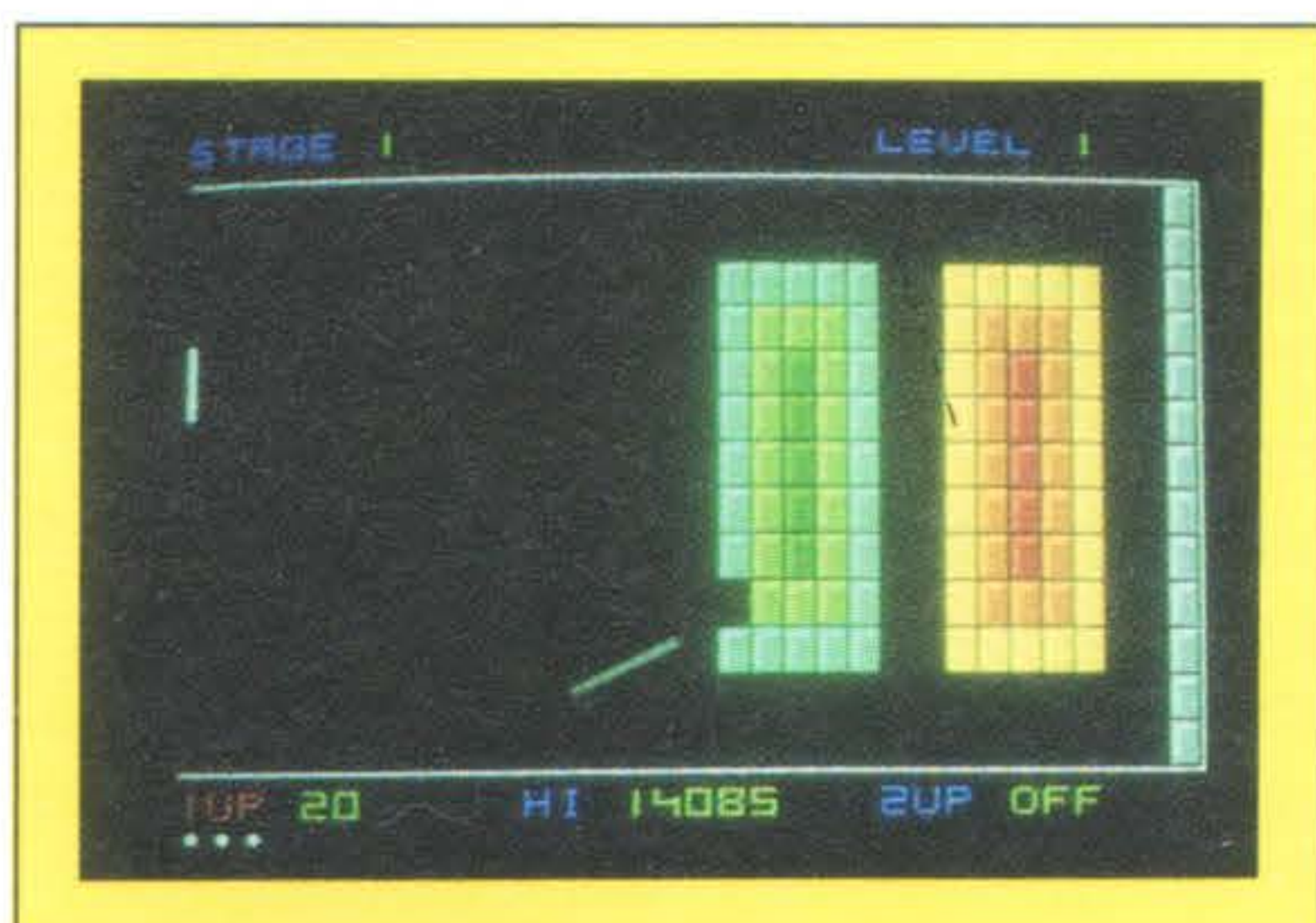
Le due linee seguenti, **FileSystem** e **DosType** (in realtà sono equivalenti, e ne basterebbe una sola) specificano se la partizione in questione può essere letta e scritta utilizzando il Fast File System, come nel nostro esempio.

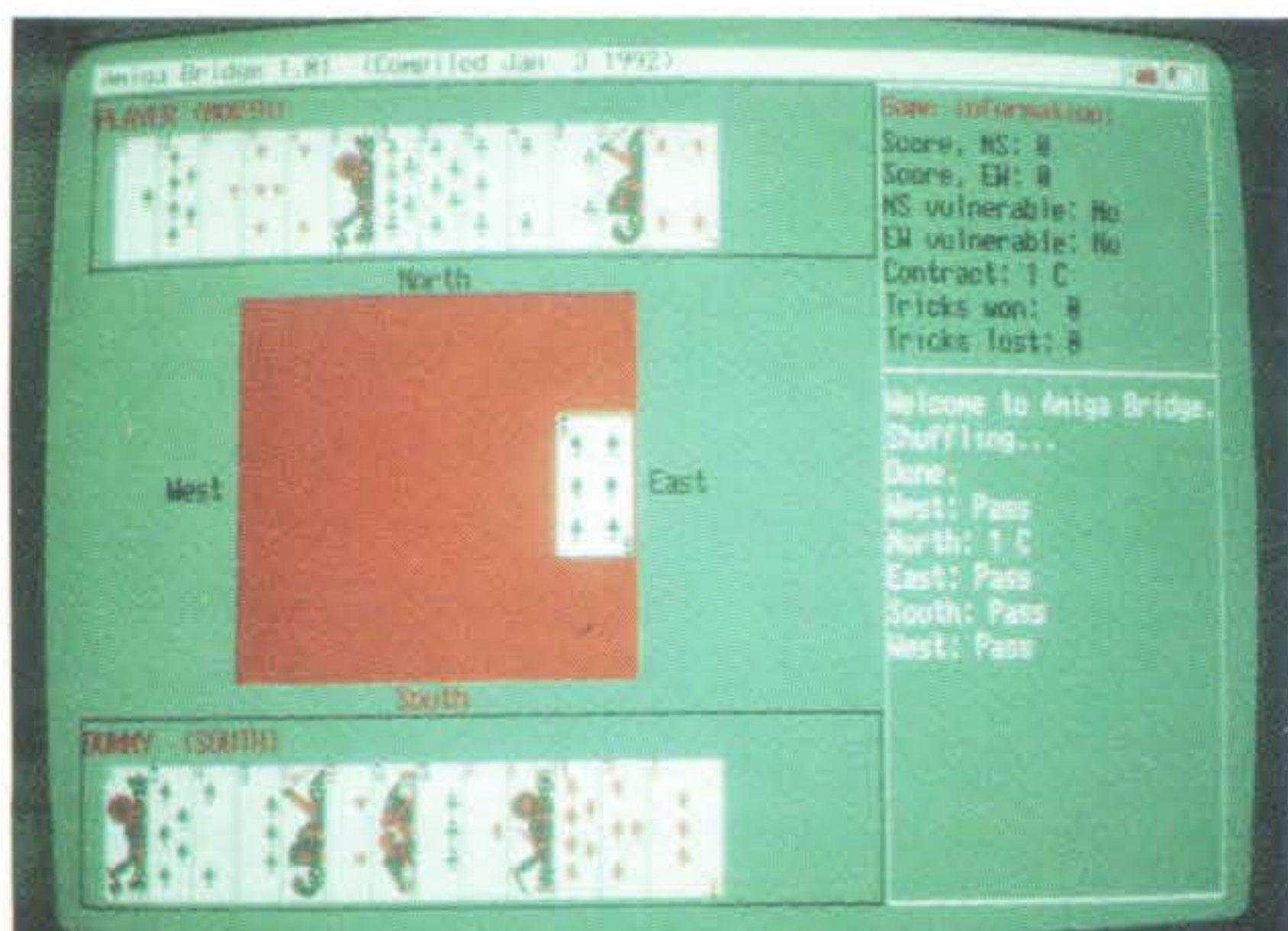
Stacksize è, infine, la quantità di memoria che il sistema operativo alloca per i dati del device driver.

Quando si effettuano modifiche alla MountList, il sistema operativo deve essere informato delle variazioni: l'operazione viene impartita da CLI o da Shell con l'istruzione **Mount dh0:** (che viene solitamente inclusa nella Startup-sequence). Senza questa istruzione il nuovo device non viene infatti visto dal sistema.

Bisogna infine ricordare che la nostra era una mountlist dimostrativa: alcuni parametri possono quindi essere omessi (nel qual caso il sistema operativo seleziona i valori di default), o apparire in diverso ordine. □

SUL DISCHETTO...

VIRUS CHECKER 6.01: L'unico modo per difendersi dai virus che infestano i dischetti di dubbia provenienza è quello di mantenersi sempre aggiornati con i più recenti programmi antivirus. Il nuovissimo «Virus Checker 6.0» individua tutti i virus attualmente noti su Amiga e permette di prevenire qualsiasi temibile infezione...
(Gira sia con il KickStart 1.3 che con il 2.0)

BUGBLASTER: Il sottosuolo del vostro giardino pullula di insetti, che si riproducono in continuazione. Riuscirete a sterminarli tutti prima che sia troppo tardi?
(Gira sia con il KickStart 1.3 che con il 2.0)

PICTURESQUE: A prima vista lo si potrebbe scambiare per «Deluxe Paint», tanta è la rassomiglianza con il celebre programma di disegno. Ma «Picturesque», oltre che per immagini pittoriche, può essere usato anche per disegnare sprite.
(Gira sia con il KickStart 1.3 che con il 2.0; richiede almeno un Mega di memoria)

PRINTICON: Stampare un testo non è mai stato così facile: basta trascinare la sua icona su quella di «PrintIcon». Un programma semplicissimo (con sorgente C incluso) che sfrutta le nuove potenzialità del WorkBench 2.0.
(Gira solo con il KickStart 2.0)

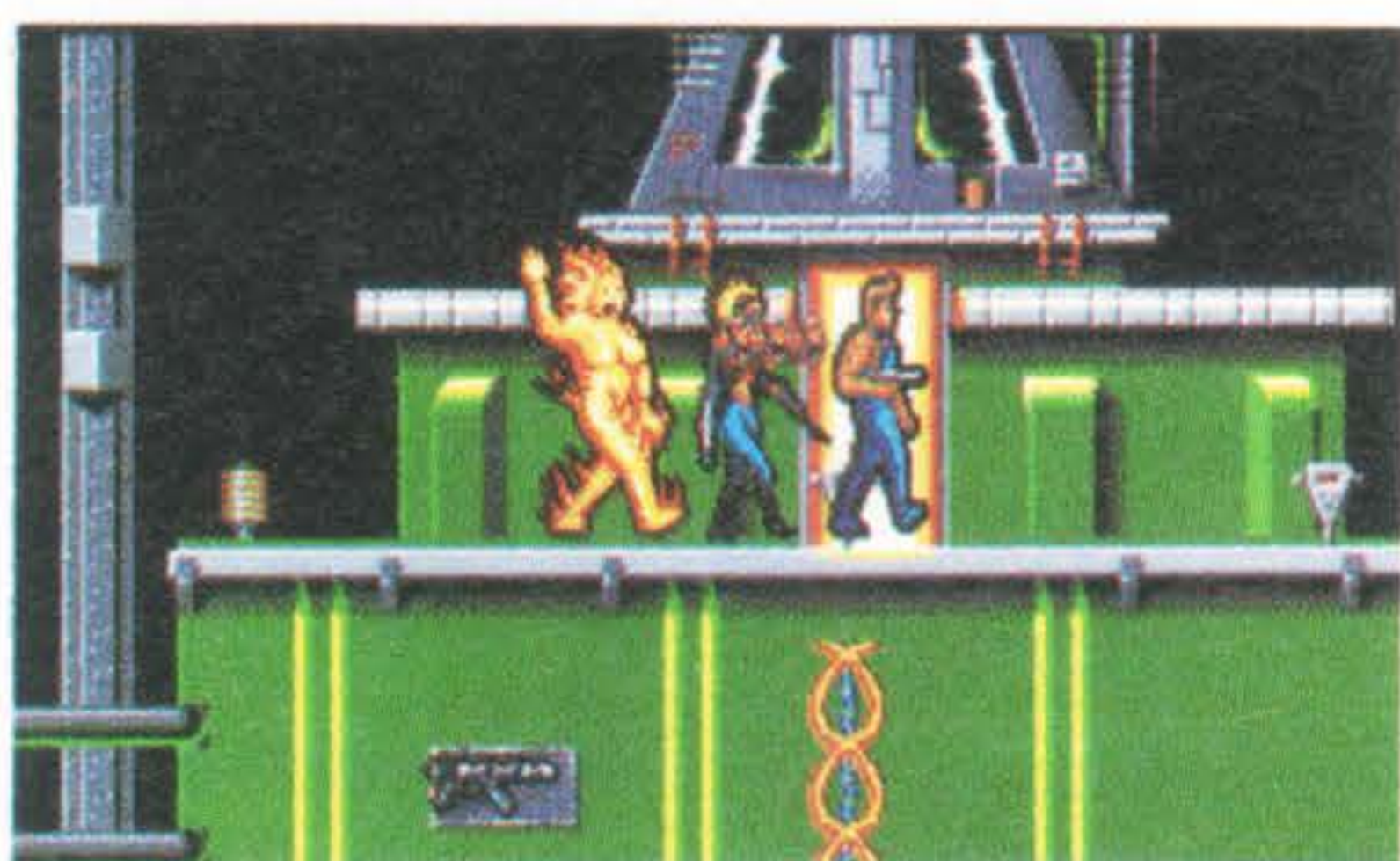
SHORTCUT: Volete lanciare qualsiasi programma con un solo tasto o tramite un comodo menu a tendina? La soluzione si chiama «ShortCut», un'utility che aggiunge menu personalizzati allo schermo del WorkBench.
(Gira sia con il KickStart 1.3 che con il 2.0)

FASTCAT: Basta un semplice cavo seriale per collegare tra loro due Amiga e potere così scambiare file, trasmettere messaggi, lanciare programmi sull'altro computer o fargli stampare file senza tenere occupato inutilmente il proprio.
(Gira sia con il KickStart 1.3 che con il 2.0)

BRIDGE: Per gli amanti dei passatempi intelligenti, una versione Amiga del più classico dei giochi di carte, che implementa fedelmente tutte le regole ed i punteggi standard.
(Gira sia con il KickStart 1.3 che con il 2.0)

POING: Di cloni di «Arkanoid» ne esistono tanti ma questo, oltre che essere incredibilmente compatto, è il più originale e divertente. Tantissimi livelli e sorprese fanno di «Poing» uno tra i migliori giochi PD mai esistiti.
(Gira sia con il KickStart 1.3 che con il 2.0)

TIV: Se trovate scomodo dover usare programmi separati per visualizzare animazioni ed immagini, o per suonare suoni campionati, con «Tiv» non avrete più problemi: in un'unica, compatta utility, c'è tutto il necessario per riprodurre file grafici o sonori in formato IFF.
(Gira sia con il KickStart 1.3 che con il 2.0)

SUGGERIMENTI E TRUCCHI VARI

Per attivare il cheat mode nascosto nel gioco «ThunderJaws» della Domark occorre avere collegati ad Amiga contemporaneamente il joystick ed il mouse. Durante il gioco, tenete premuti il tasto sinistro del mouse ed il tasto «S» e, quando volete passare al livello successivo, premete anche il tasto destro del mouse.

Anche il classico gioco «Defender II» di Jeff Minter contiene un cheat mode. Digitando queste parole potete attivare alcune funzioni speciali:
GOATY - Diventate invulnerabili
RAVEN - Permette di saltare al livello successivo premendo il tasto N, o di attivare il pilota automatico premendo D.
INCAS - Aumenta la potenza del laser.
ANDES - Ripristina l'energia ogni volta che usate il laser.

Come se non bastasse, ecco alcune delle password per accedere ai vari livelli di gioco:
1) START 5) FLOYD 9) FURRY 13) BEAST 17) LEMAC 21) ZIPPO 25) LASER 29) DEFAD 33) MAGOG 37) FUNKY 41) DONKY 49) KANJI 53) IRATA 57) NEURO 61) STOAT

In «Metal Master», gioco di pugilato futuristico della francese Infogrames, premendo F4 durante un combattimento paralizzerete il robot avversario e potrete facilmente averne ragione.

Digitate il nome «SHAFT» nella tabella degli high scores nel gioco «Dominator» per avere vite infinite.

Ecco le password per tutti i livelli del rompicapo giapponese «Gem-X»:

B - EARTHIAN C - KENICHI
D - INOKHUMA E - BURAI
F - BADMAN G - NETWORK
H - YOKOHAMA I - EXACT
J - X68000 K - TURRICAN
L - REDMOON M - CAMPAIGN
N - MAGAMANN O - SYVALION
P - FMTOWNS Q - CHIERIE
R - GAMERION S - ZAWAS

Anche il simpatico «Globulus», un arcade ispirato al classico «Q-Bert», contiene un cheat mode: per attivarlo, alla richiesta della password per accedere ai livelli di gioco, inserite questo codice: «Zvmo! EG».

Attenzione ad inserire correttamente le maiuscole e le minuscole così come le vedete qui.

I Role Playing Game, più familiarmente noti sotto l'abbreviazione RPG, sono una categoria di gioco che sta avendo sempre più successo. «Crystals of Arborea», della Palace Software, è tra gli ultimi usciti.

Nella schermata con i personaggi selezionate Jarel, clickate sull'icona con le bottiglie e premete Ctrl-V per riportare ai valori massimi tutti i punti di energia del vostro personaggio.

Digitando la frase «IN A GARDEN» mentre appare la schermata iniziale nel gioco «Chase HQ II» si attiva un cheat mode.

Durante il gioco potrete poi premere il tasto T per ottenere più tempo.

Pare inoltre che tenendo premuti contemporaneamente i tasti H e F5 durante il caricamento del gioco, gli ostacoli lungo la strada non compaiano durante la corsa.

Nell'avventura «Enchanted Lands» digitate la frase «TCB RULES FOREVER» durante la schermata introduttiva, e lo schermo lampeggerà.

A questo punto potrete premere il tasto F3 per accedere ad un editor interno per variare i parametri del programma; premendo invece durante il gioco F2 seguito da uno spazio verrete trasportati direttamente davanti al guardiano di fine livello.

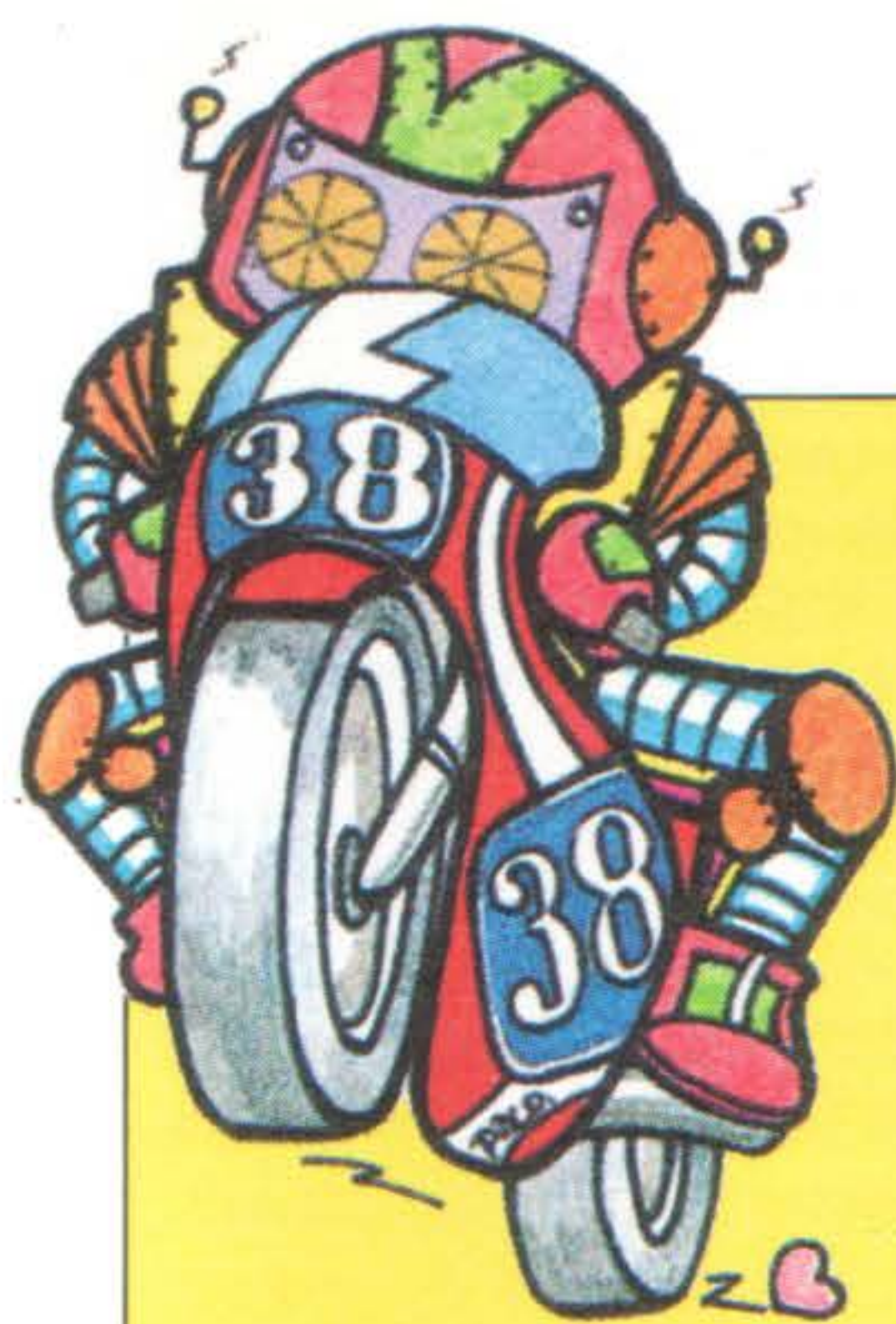
Se siete tra i fortunati possessori di una stampante laser, per la precisione di un modello Hewlett Packard LaserJet, avrete probabilmente già riscontrato uno sgradevole inconveniente causato da una limitazione del driver di stampa HP LaserJet fornito a corredo del WorkBench: la lunghezza massima della pagina stampata risulta essere 10 pollici.

Questo rappresenta un problema quando occorre produrre stampe in formato A4, la cui lunghezza è superiore ai 10 pollici gestiti dal driver. La soluzione è a portata di mano, disponendo di un qualsiasi file editor come «NewZap» o «FileMaster».

È sufficiente localizzare, all'interno del file, la sequenza di caratteri esadecimali E5 80 D0 81 e modificarla in questo modo: E7 80 90 81. La stringa in questione si trova nel settore 10 del driver, offset 15E. In questo modo la lunghezza massima della pagina viene alterata in 14 pollici, risolvendo tranquillamente l'inconveniente. Conviene ovviamente lavorare su di una copia del driver, per evitare sgradevoli sorprese in caso di errore.

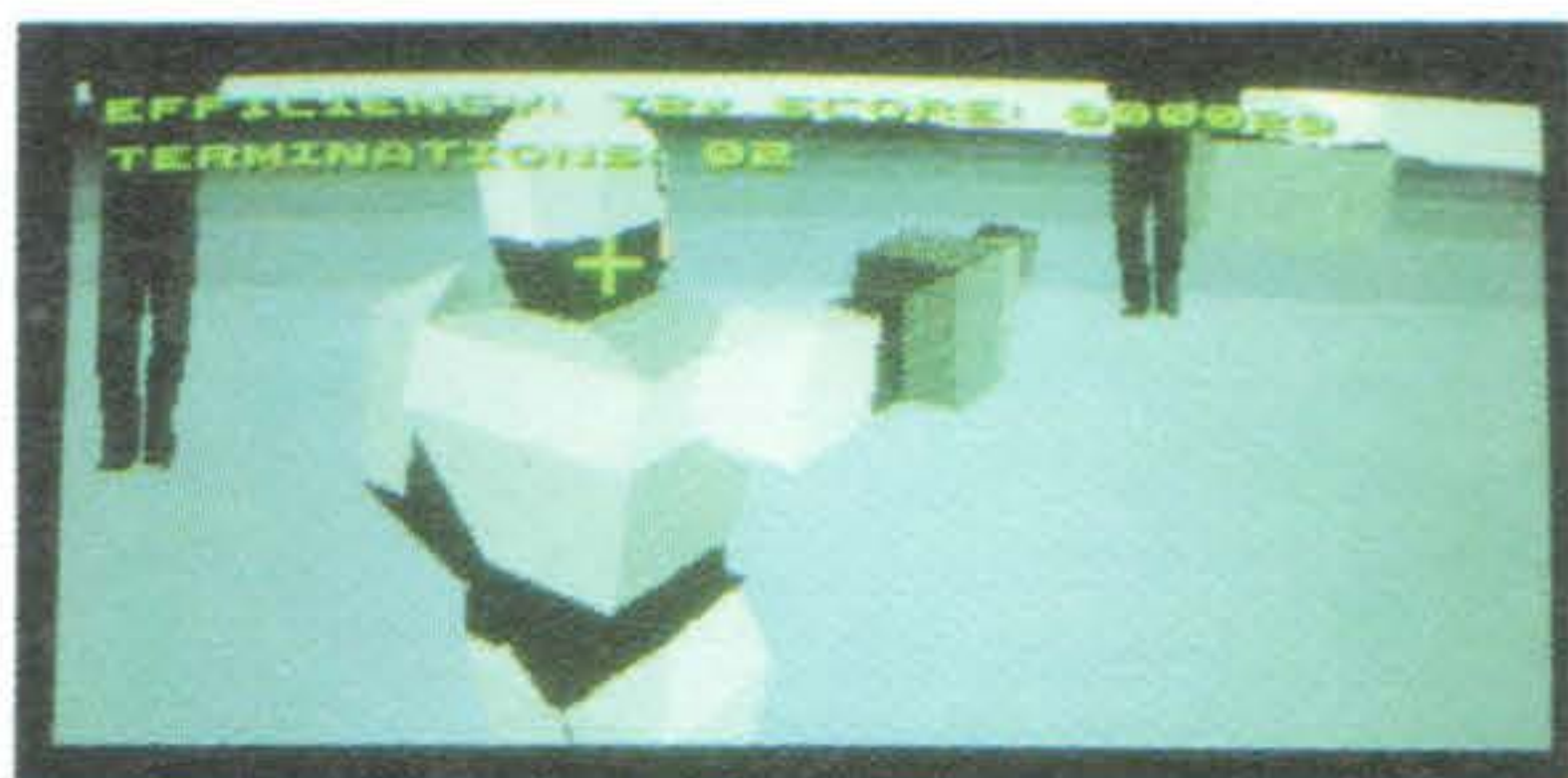
Per ottenere vite infinite con «Turrican II», andate al menu di selezione della musica prima di iniziare il gioco e digitate il numero 42 seguito da due pressioni del testo Esc.

Nel gioco «Dragon Ninja» provate a digitare la parola «TERRIFIC» durante la partita. Premendo il tasto «L» passerete al livello successivo, mentre con F3 sarete dotati di vite infinite. Anche nel gioco «Elf» si attiva un cheat mode: la parola da digitare è «CHOROPOO».

Conversione di un popolarissimo gioco per Commodore 64, «Armalyte» è tra i più impegnativi shoot'em up attualmente in circolazione per Amiga. Per attivare il cheat Mode e disporre di vite infinite, inserite la pausa e digitate la parola «DELTA3». Se non avete commesso errori, vedrete cambiare il colore del pannello con il punteggio nella parte inferiore dello schermo e, una volta tolta la pausa, sarete invulnerabili.

Software Express

di Marco Brovelli

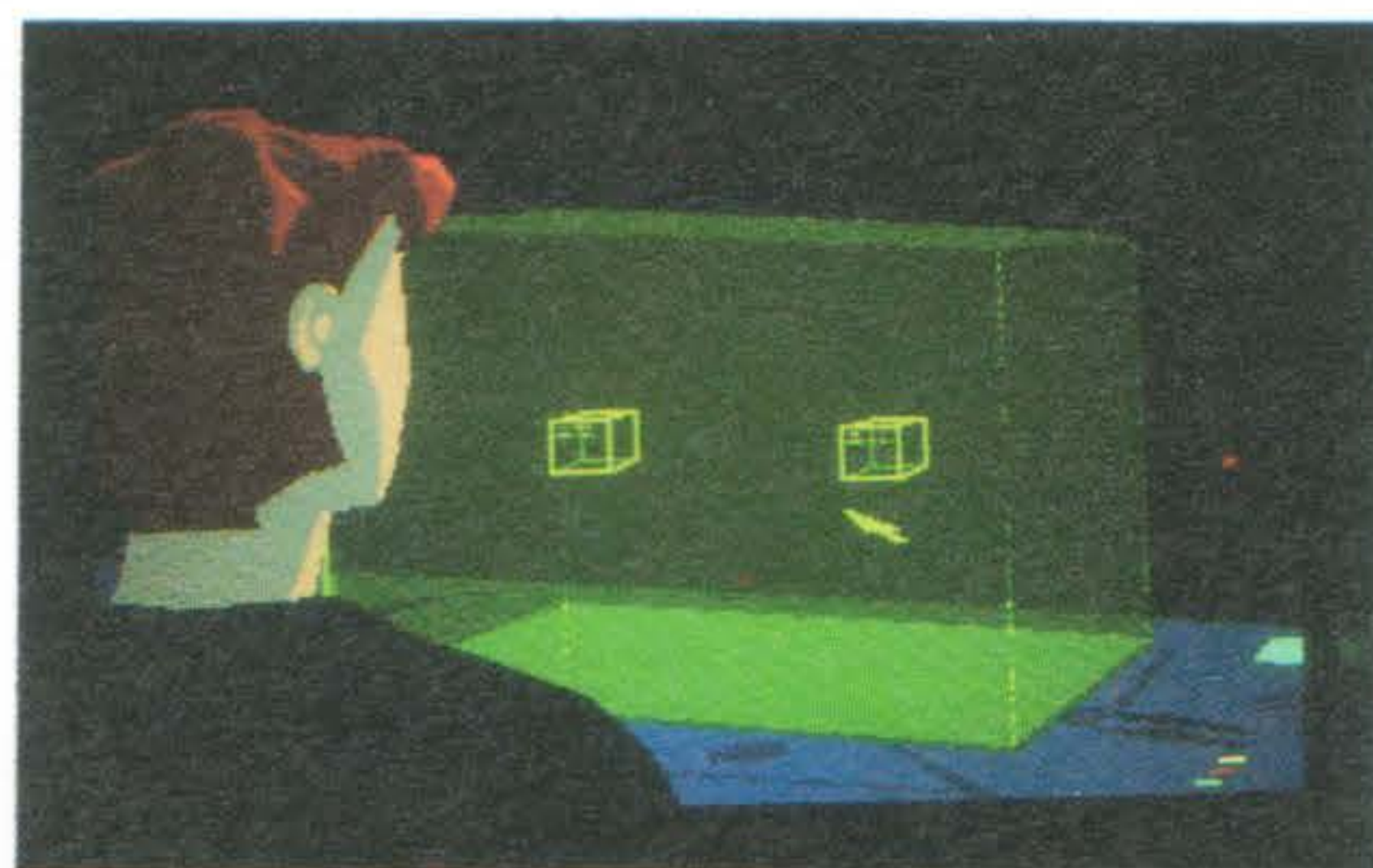
ROBOCOP 3

Venghino, signore e signori, ad ammirare «Robocop 3», il primo caso di gioco ispirato da un film che ancora non esiste.
La storia è drammaticamente verosimile: Delta City (ovvero Detroit, un tempo fiorente centro dell'industria automobilistica) ha conosciuto la disoccupazione a causa della concorrenza dei giapponesi. Quasi tutti gli abitanti hanno perso la casa, e la città è ora in mano a pericolosi gruppi di malviventi.
L'organizzazione deputata al mantenimento dell'ordine, l'OCP, ha instaurato un clima di terrore grazie alle molte azioni contro i teppisti, costate molte vittime fra i civili.
Robocop, naturalmente programmato per non uccidere gli innocenti, ha abbandonato l'OCP che, per ritorsione, lo fa passare per un fiancheggiatore dei criminali.
Proprio una bella situazione, non c'è che dire! Inutile specificare che nei panni (o meglio, nelle corazze) del robot ci sarete voi.

Le missioni che sarete chiamati a compiere sono numerose: dall'inseguimento di un camion carico di armi, alla liberazione di ostaggi nei luoghi più disparati; dal combattimento aereo (in questo episodio infatti Roby può anche volare) al duello finale con altri robot dagli ideali meno alti dei vostri.
Tutte le fasi di gioco sono realizzate con grafica vettoriale tridimensionale davvero fluida; potete visualizzare la scena in prospettiva soggettiva (così come la inquadrerebbe il robot), o da svariati punti di vista.
Effetti sonori piuttosto azzeccati (non c'è una vera e propria musica) ed un'ottima rispondenza tra movimenti del mouse ed azioni del personaggio portano «Robocop 3» decisamente al di sopra della media.
Tutti gli aggiornamenti sulla situazione a Delta City sono forniti in tempo reale dal telegiornale i cui anchormen, tuttavia, non si riveleranno esenti dall'influenza di una potente multinazionale giapponese...

MICROPROSE GOLF

L'ultimo prodotto della Microprose, software house americana nota prevalentemente per le sue simulazioni di volo, è l'ennesima variante sul tema del gioco del golf, uno sport le cui versioni a 16 bit sono superate (in numero) soltanto da quelle automobilistiche.
Fortunatamente questo «Microprose Golf» è il migliore tra tutti i simulatori di golf finora apparsi su Amiga: la velocità, la qualità della grafica, la dovizia di opzioni e, soprattutto, il realismo del gioco sono superiori a quelli di tutti i suoi concorrenti.
Potete variare tutti i parametri possibili ed immaginabili: dalla posizione dei piedi sul terreno a quella del supporto che regge la pallina oltre che, naturalmente, al tipo di mazza, all'angolatura, ed alla forza con la quale il colpo viene effettuato.
La grafica è eccellente: oltre che la tradizionale immagine del *green* visto da dietro le spalle del giocatore, si può richiamare una vista prospettica della buca e ruotarla a piacimento. La ripresa più spettacolare consiste nella possibilità di seguire il punto di vista della pallina mentre vola sul terreno, con una velocità ed una fluidità di movimento davvero notevoli.
Il programma comprende già sei diversi campi (ma è lecito aspettarsi l'uscita di data-disk con percorsi aggiuntivi) sui quali misurarsi con altri giocatori, fino a quattro, che possono essere umani o controllati dal computer.
Se siete amanti del golf, questo è il programma che fa per voi; ed anche se non lo siete, vale la pena comunque di provarlo, perché è tra i migliori giochi Amiga degli ultimi mesi.

ANOTHER WORLD

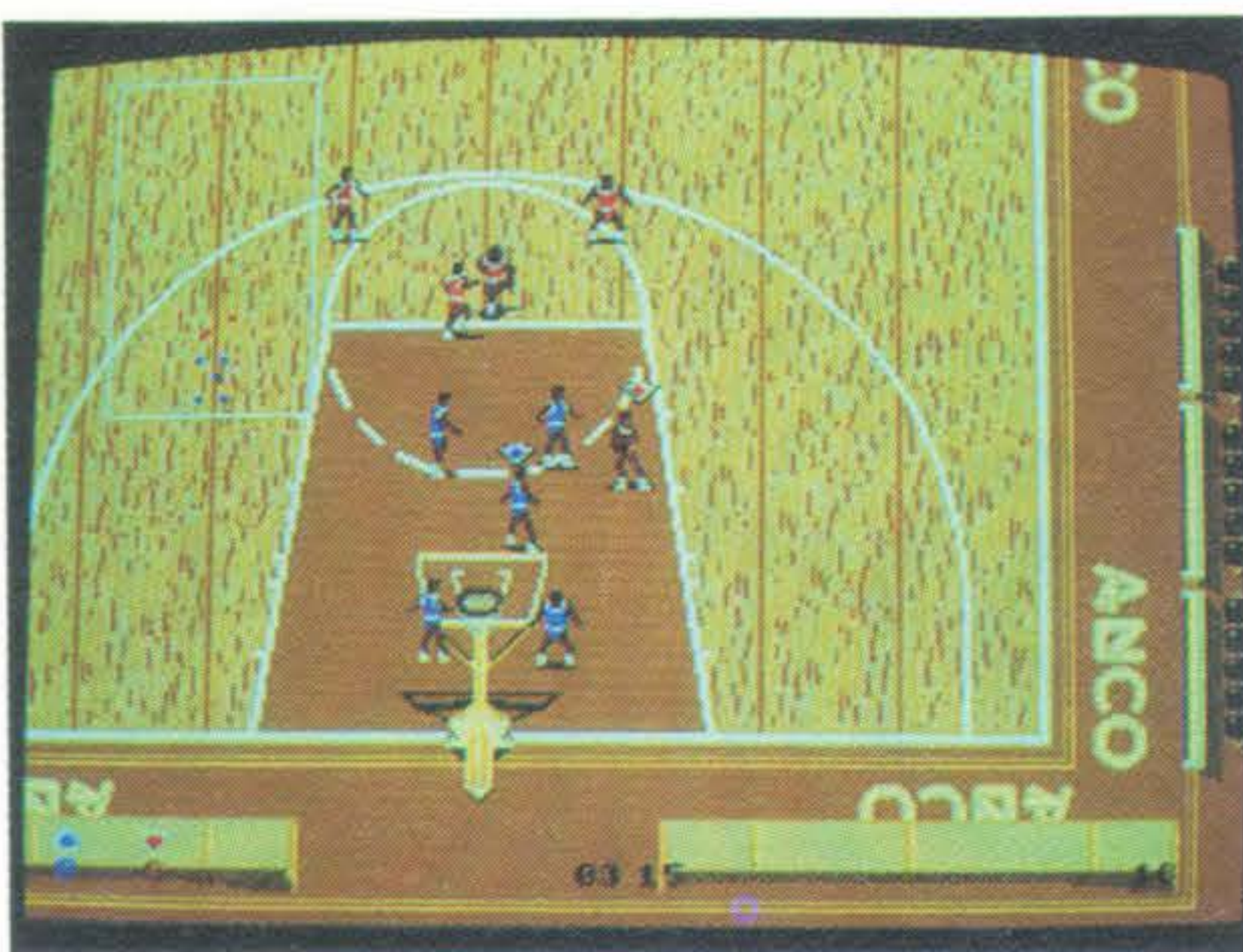
Sorprendente è l'aggettivo giusto per definire l'ultimo parto a 16 bit della Delphine, la software house francese che da pochi anni si è affacciata sul mercato Amiga con alcuni titoli di ottimo livello. Questo «Another World» sfrutta le routine grafiche introdotte per la prima volta con l'avventura «Cruise for a Corpse», un giallo alla Agatha Christe, e le ripropone in una cornice fantascientifica con l'aggiunta di elementi arcade.
Il risultato è questo «Another World», uno tra i pochissimi giochi che possono fregiarsi con diritto della definizione, tanto abusata, di «film interattivo».
Il vostro ruolo è quello di un giovane scienziato che, in una notte buia e tempestosa, conduce un solitario esperimento nucleare con un acceleratore di particelle. L'imprevisto, sotto forma di fulmine, fa fallire tragicamente la prova, ottenendo lo sgradevole effetto collaterale di teletrasportarvi nell'«altro mondo» dal quale il programma prende il titolo. Vi ritroverete quindi su di un misterioso pianeta popolato da animali mostruosi e da bizzarri alieni che non troveranno di meglio che rinchiudervi nelle loro prigioni.
Riuscirete a sopravvivere ed a ritornare sulla Terra? Dipende dalla vostra bravura con i platform game in stile «Prince of Persia»: poiché «Another World», sotto sotto, non è altro che un gioco d'azione mascherato da avventura interattiva grazie all'aggiunta di spettacolari sequenze animate.
Non lasciatevi comunque scoraggiare da questa affermazione: giochi di questo livello se ne vedono raramente. La qualità della grafica è, come abbiamo detto, superlativa; il sonoro e le animazioni non sono da meno; la giocabilità è più che buona e, dulcis in fundo, la bizzarra trama è originale quanto basta a fare di «Another World» un classico che resterà per sempre negli annali della storia di Amiga.

BIG RUN

Quante possibilità ha un gioco di corsa automobilistica di superare il primato di eccellenza attualmente detenuto dal giocabilissimo «Lotus Turbo Esprit Challenge II» della Gremlin? Molto poche,

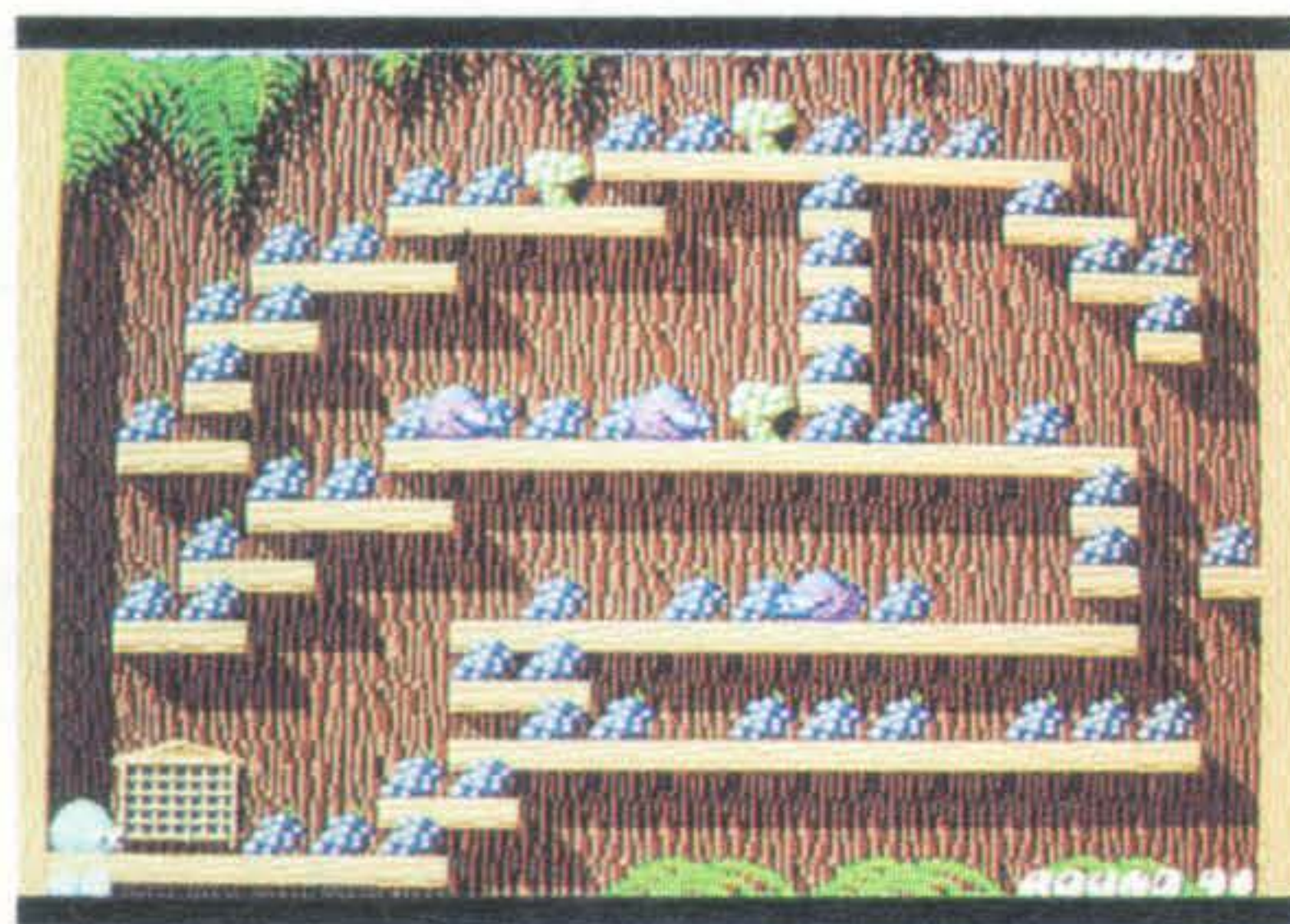
Software Express

in verità: la conferma è data da questo «Big Run», una conversione decisamente poco convinta dell'omonimo gioco arcade della Jaleco.
Nelle intenzioni dei programmatori, «Big Run» vorrebbe offrire la possibilità di vivere le emozioni dei piloti che gareggiano nella Parigi-Dakar; in realtà, l'unica connessione con la celebre competizione sta nei nomi delle tappe e nell'ambientazione desertica, la cui monotonia grafica non contribuisce certo ad appagare gli occhi del giocatore.
I livelli di gioco sono sei, tutti molto simili, da percorrere secondo le regole standard dei videogame automobilistici: arrivare al traguardo prima che scada il cronometro per ottenere un bonus di secondi extra, oppure lasciar scadere il tempo e veder apparire la tragica scritta «Game Over». Il tutto condito con il solito assortimento di auto avversarie da superare, tutte tragicamente uguali, di dossi e fiumi (nel deserto!) sui quali saltare, di macigni ed alberi ai margini della strada da scansare.
Meglio sorvolare sul realismo della competizione, dato che il contachilometri della nostra vettura, simile più ad un maggiolino Volkswagen che ad una macchina da rally, arriva tranquillamente alla fantascientifica velocità di 320 Km orari, dando al giocatore la stessa sensazione di verosimiglianza di un arcade spaziale.
Se questa descrizione non sembra invitante è perché il gioco non è molto più divertente di una seduta dal dentista. «Big Run» è insomma uno di quei giochi sulla cui confezione andrebbe apposto un bollino con l'avvertenza «Solo per masochisti».

TIP OFF

Chi non ha mai visto «Kick Off», il famoso simulatore di calcio, entrato ormai nella breve storia del software per il riuscitissimo connubio fra strategia e giocabilità?
La Anco prova a bissare il successo del suo programma più noto con «Tip Off», un prodotto analogo ma indirizzato ai numerosi estimatori del basket.
Il risultato non è male: dal punto di vista della gestione della squadra ci sono stati dei miglioramenti, mentre l'azione vera e propria non raggiunge, a nostro parere, i livelli del predecessore.
Potete stabilire ogni minimo aspetto del vostro team: dalla composizione alla disposizione ed ai movimenti dei giocatori in campo, al campionato in cui iscriverla (internazionale, nazionale, locale o amatori,

corrispondenti ad altrettanti livelli di difficoltà). Giungiamo così alle partite vere e proprie: il campo è inquadrato dall'alto: voi guidate un giocatore, mentre gli altri si comporteranno (si spera) secondo gli schemi da voi stabiliti in precedenza. In questo modo non potrete più dare la colpa di una sconfitta al computer, dato che direttamente o indirettamente siete voi a pilotare l'intera squadra.
Lodevole lo sforzo di internazionalizzazione della Anco, che ha realizzato il gioco in quattro lingue compreso l'italiano: peccato che le traduzioni non siano migliorate dai tempi di «Kick Off» (ricordate il «Calcio d'angelo»? Preparatevi a «Squadra colori», «Inserite i dati disco» ed altro!).
Si tratta comunque di un titolo che, ne siamo sicuri, non mancherà sugli scaffali degli amanti delle simulazioni sportive.

FUZZBALL

Non sappiamo esattamente che razza di animale sia un *fuzzball*, il protagonista dell'omonimo gioco della System 3: d'altronde in un videogame è lecito aspettarsi di tutto, anche di impersonare il ruolo di una buffa palla di pelo con due occhioni sbarrati, che deve percorrere (pardon: rimbalzare in) cinquanta livelli e raccogliere tutti i gioielli ed i bonus in essi contenuti.
Trattandosi di un platform game, l'impresa non è esattamente una passeggiata: ogni schermo è infestato da un numero variabile di vostri simili, che manifestano diversi livelli di ostilità nei vostri confronti ed il cui contatto, naturalmente, è letale. È il colore dei fuzzball nemici a determinarne la velocità ed il grado di pericolosità: i più innocui sono verdi.
Potete sparare addosso ai fuzzball nemici, fino a farli trasformare in una minuscola palla che rimbalza brevemente sullo schermo: una volta colpito un fuzzball dovete però neutralizzarlo definitivamente andandolo a toccare mentre si trova ancora in questo stato. In caso contrario, tornerà a tramutarsi in un fuzzball, questa volta però

di un colore più pericoloso (da verde a viola, a nero ed infine a rosso).
«Fuzzball» è un gioco molto tradizionale: niente di sensazionale sotto il profilo grafico o sonoro, ma di buon livello per quanto riguarda la giocabilità ed il divertimento. Cinquanta livelli possono apparire pochi, ma sono impegnativi al punto da non far sentire la mancanza di altri schermi.
L'appunto maggiore che si può muovere riguarda proprio la difficoltà, che ai meno esperti potrà apparire eccessiva e che finisce con lo scoraggiare facilmente i meno tenaci.

LEANDER

Siamo alle solite! Un'altra fanciulla rapita da sottrarre alle grinfie del malvagio demone di turno.
In attesa che qualche software house decida di introdurre qualche nuova variante alla formula consumata del principe azzurro che deve salvare la sua bella in pericolo, dobbiamo accontentarci di queste trame ritrite da fotoromanzo. Per fortuna, a volte, esse fungono da contorno a giochi di buon livello.

È il caso dell'ultimo nato della Psygnosis, un nome che è quasi una garanzia. «Leander» è un platform game nello stile di «Barbarian», nel quale il protagonista è armato di un lungo spadone e deve superare orde di mostri più o meno giganteschi, saltare qua e là su piattaforme semoventi ed evitare altre insidie che infestano i livelli del gioco. Non mancano (come potrebbero?) i tradizionali forzieri che contengono denaro, vite extra ed altri bonus indispensabili per la sopravvivenza di qualsiasi guerriero computerizzato.
Il look del protagonista è decisamente orientale ed ispirato ai *manga* giapponesi: l'animazione dello sprite principale è molto curata, i controlli sono precisi e fluidi, contribuendo a rendere «Leander» molto giocabile.
A differenza di diversi arcade adventure, come «Shadow of the Beast», in cui il giocatore viene assalito da fiumane di zombi che gli si avventano addosso da ogni direzione, «Leander» è più riflessivo e meno frenetico: questo dà il tempo di pianificare le proprie mosse e di calcolare meglio le conseguenze di un salto o la tattica per fare a fettine il super-mostro di turno.
Un prodotto molto curato, insomma, al quale tuttavia un po' di originalità in più non avrebbe certo fatto male.

BIRDS OF PREY

I simulatori di volo disponibili per Amiga generalmente ricadono in due categorie ben

precise: quelli che pongono l'accento sul realismo della simulazione, per dare al giocatore l'impressione di sedere veramente dietro alla cloche di un caccia da combattimento o di un aereo di linea, e quelli (la maggior parte) che preferiscono dare spazio alla giocabilità e riducono la quantità di comandi disponibili e la difficoltà nel pilotare il velivolo a vantaggio di una maggiore semplicità di utilizzo e fluidità dei movimenti.
In attesa che esca il simulatore di volo perfetto, in grado di fornire l'ideale connubio di queste due virtù, accontentiamoci di «Birds of Prey», l'ultimo nato della Electronic Arts.
Dodici tipi di missione (con numerose varianti) e quaranta diversi aerei da pilotare donano al gioco uno spessore che pochissimi simulatori possono vantare, rendendo quasi impossibile annoiarsi.
La grafica, a vettori solidi, è al livello standard di questo genere di gioco e risente dell'inconveniente classico di tutti i programmi che comprendono animazione 3D: la lentezza. È comunque possibile selezionare la definizione dei dettagli degli oggetti visualizzanti: al livello più alto, il panorama circostante ed i velivoli nemici sono più definiti, a scapito di una minore velocità nell'aggiornamento dello schermo, e viceversa.

«Birds of Prey» ricorda molto il vecchissimo «F/A 18 Interceptor», sempre della Electronic Arts: stessa predilezione per l'aspetto giocoso della simulazione e stesso divertimento nell'eseguire acrobazie a bordo del proprio caccia.
Inutile dire che, tecnicamente, gli anni che separano i due programmi si fanno sentire: la dovizia di comandi, di missioni e di dettagli fanno di «Birds of Prey» il miglior gioco di simulazione aerea del momento.

MEGA TWINS

Il vostro paese è stato invaso da un'entità maligna con il dovuto contorno di ributtanti mostriciattoli. Anche questa volta,

casualmente, siete stati scelti proprio voi per liberare la patria. Non attardatevi a meditare se si tratti di fortuna o di sfortuna, perché non avete un secondo da perdere. Sia la terra che l'acqua che l'aria sono infestate: nell'ordine che preferite, dovrete liberarle tutte.
Il meccanismo di gioco, i più perspicaci lo avranno già intuito, è quello tipico dei platform game; in particolare, «Mega Twins» presenta spiccate affinità con «Rodland», con il quale ha in comune la grafica coloratissima e bordata in nero, tipica dei cartoni animati.
Al termine di ciascun livello si trova il tradizionale mostro più cattivo e coriaceo di tutti gli altri; considerato che tutto ciò che avete a disposizione è una spada, comprenderete quanto questo incontro sia poco piacevole.
Se riuscirete ad eliminare il mostro, farete conoscenza con la divinità corrispondente all'elemento nel quale avrete combattuto (ci permettiamo di osservare come la dea della Terra sfoggi generosamente un paio di gambe da cardiopalma), che ci regalerà un'arma da utilizzare durante lo scontro finale. Quest'ultimo avverrà soltanto quando avremo terminato tutti e tre i livelli; per avere qualche chance di vittoria, è caldamente consigliabile essere in due.
Come dite? Non l'avevamo detto? Già, ma si capiva dal titolo: in «Mega Twins» è possibile giocare contemporaneamente in due, dividendosi naturalmente le vite a disposizione.

La grafica, come si è detto, è gradevole. Vi sono dei momenti, nel gioco, in cui desiderereste un'azione più rapida (vuoi per la lentezza dei movimenti dei personaggi, vuoi per la presenza di alcuni nemici che vanno colpiti molte decine di volte prima di poter proseguire). Per dirla in breve: carino, ma poco originale.

nel prossimo fascicolo di

- IMAGINE 2.0 TUTORIAL
- SUPER JAM!
- VIDEO DIRECTOR
- WORD PROCESSING

e mille altre cose ancora...

ABBONARSI CONVIENE!
SOLO LIRE 135.000
PER 11 FASCICOLI E 11 DISCHETTI DIRETTAMENTE A CASA TUA OGNI MESE
CLICKA SU AMIGA BYTE!
Cosa aspetti?
AMIGA
IN REGALO 2 SUPERDISCHI:
TOP GAME
TOP UTILITY
Oppure, a scelta, due dischetti della nostra raccolta di software di Pubblico Dominio (specificare i codici dei dischi desiderati sul vaglia).
Puoi abbonarti anche alla sola rivista (senza disco): Lire 85.000 (1 disco omaggio a scelta).
L'abbonamento a 5 fascicoli completi di dischetto costa solo Lire 65.000 (1 disco omaggio a scelta).
Per abbonarti, invia vaglia postale ordinario ad Amiga Byte, c.so Vitt. Emanuele 15, 20122 Milano.
Indica sulla parte destra del vaglia, nello spazio delle comunicazioni del mittente, che desideri abbonarti ad Amiga Byte, il nome o il codice dei dischi omaggio che preferisci, ed i tuoi dati in stampatello, completi.